UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Luiz Felipe da Silva

REDUÇÃO DE RUÍDO EM SINAIS DE VOZ UTILIZANDO UMA FUNÇÃO DE GANHOS ADAPTATIVA PARA O FILTRO DE WIENER

Florianópolis

2011

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Luiz Felipe da Silva

REDUÇÃO DE RUÍDO EM SINAIS DE VOZ UTILIZANDO UMA FUNÇÃO DE GANHOS ADAPTATIVA PARA O FILTRO DE WIENER

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina para a obtenção do grau de Mestre em Engenharia Elétrica.
Orientador: José Carlos Moreira Bermudez

Florianópolis

2011

Luiz Felipe da Silva

REDUÇÃO DE RUÍDO EM SINAIS DE VOZ UTILIZANDO UMA FUNÇÃO DE GANHOS ADAPTATIVA PARA O FILTRO DE WIENER

Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.
Florianópolis, 16 de dezembro de 2011.

_____

Patrick Kuo Peng, Dr.
Coordenador do Curso

Banca Examinadora:

_____

José Carlos Moreira Bermudez, PhD
Orientador

_____

Joceli Mayer, Ph.D

_____

Márcio H. Costa, Dr.

_____

Carlos Aurélio Faria da Rocha, Dr.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, ao meu irmão e à minha namorada por todo o apoio durante o decorrer do curso de Mestrado.

Agradeço aos professores Márcio Holsbach Costa e Joceli Mayer pelo suporte e oportunidade de integrar o grupo de alunos de pós-graduação do LPDS. Agradeço principalmente meu orientador, José Carlos Moreira Bermudez, pelo conhecimento transferido e pelo constante apoio nas dificuldades encontradas. Agradeço não somente por sua orientação acadêmica, mas também por todos os conselhos dados que foram muito importantes para meu desenvolvimento pessoal.

Agradeço aos colegas do laboratório LPDS Daniel Matos Montezano, Osmando Pereira Júnior. Agradeço também aos colegas Wemerson Delcio Parreira e Marcos Hideo Maruo por todo apoio e companheirismo no decorrer do curso. Agradeço principalmente à minha colega Renata Coelho Borges pela amizade e por toda o auxílio prestado no momentos que precisei.

Por fim, agradeço à CAPES e à FAPEU pelo suporte financeiro.

RESUMO

Muitas técnicas de redução de ruído, especialmente a filtragem de Wiener, sofrem com a introdução de ruído musical e distorção do sinal de voz em SNRs baixas devido às suas funções ganho rígidas. Neste trabalho propomos uma modificação do filtro de Wiener paramétrico para enfatizar as contribuições espectrais nas regiões do espectro que são importantes para inteligibilidade. Isto é feito definindo um parâmetro adaptativo que é uma função do *pitch*. Medidas objetivas e testes estatísticos são usados para avaliar a qualidade subjetiva e inteligibilidade do sinal de voz. Os resultados indicam que o algoritmo proposto resulta na melhora da inteligibilidade e redução do ruído musical do sinal de voz processado, comparado com o filtro de Wiener convencional.

Palavras-chave: Filtro de Wiener paramétrico, teste estatístico, PESQ, CSII, pitch, inteligibilidade, ruído musical.

ABSTRACT

Many existing speech enhancement techniques, especially Wiener filtering, suffer from introducing annoying musical noise and speech distortion in low SNR due to their rigid gain functions. In this work we propose a modification to the parametric Wiener filter that emphasizes the spectral contributions in spectral regions which are important for intelligibility. This is done by defining an adaptive parameter that is a function of the pitch. Objective measures and statistical tests are used to assess subjective speech quality and intelligibility. The results indicate that the proposed algorithm results in speech intelligibility improvement and in musical noise reduction, as compared to the conventional Wiener filter.

Keywords: Parametric Wiener filtering, statistical test, PESQ, CSII, pitch, intelligibility, musical noise.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AR	Auto-regressivo.
AWP	Algoritmo de Wiener proposto.
CR	Conjunto de Ruídos.
CSII	Coherence-based speech intelligibility index.
CSV	**Comma-separated values**, em português Valores Separados por Vírgula.
CT	Conjunto de Treinamento.
CV	Conjunto de Validação.
DEP	Densidade Espectral de Potência.
DFT	Transformada Discreta de Fourier.
DTFT	Discre-time Fourier Trasform.
EVD	Eigenvalue Decomposition.
FDP	Função de Densidade de Probabilidade.
FIR	Filtro de resposta finita.
IIR	Infinite Impulse Response Filter.
KLT	Transformação de Karhunen-Loève.
LMMSE	Linear Minimum Mean-Square Estimator.
LSE	Least Square Estimator.
PESQ	Perceptual Evaluation of Speech Quality.
SDCE	Spectral-Domain-Constrained Estimator.
SNR	Signal to Noise Ratio.
SVD	Singular Value Decomposition.
TDCE	Time-Domain-Constrained Estimator.
VAD	Voice Activity Dectector.

LISTA DE SÍMBOLOS

n	Índice de tempo
$x(n)$	Sinal limpo
$w(n)$	Ruído aditivo
$y(n)$	Sinal de entrada ruidoso
$d(n)$	Resposta desejada
$e(n)$	Erro de estimação
$d(n)$	Resposta desejada
$\hat{d}(n)$	Estimativa da resposta desejada
$h(n)$	Resposta ao impulso do filtro LTI de duração infinita
$*$	Convolução linear
$h(n)$	Filtro LTI
$H(\omega)$	DTFT de $h(n)$
$Y(\omega)$	DTFT de $y(n)$
$\hat{D}(\omega)$	DTFT de $\hat{d}(n)$
M	Número de pontos da DFT
$\psi(\omega_k)$	DFT de $e(n)$
$E\{\cdot\}$	Operador de valor esperado
a^*	Complexo conjugado de a
$P_{yy}(\omega_k)$	Espectro de potência do sinal de entrada $y(n)$
$P_{dy}(\omega_k)$	Espectro de potência cruzado entre $y(n)$ e y(n)
$W(\omega_k)$	DFT de w(n)
$P_{xx}(\omega_k)$	Espectro de potência do sinal de entrada $x(n)$
$P_{ww}(\omega_k)$	Espectro de potência do sinal de ruidoso $w(n)$
$\hat{x}(n)$	Estimativa do sinal limpo (filtrado) $x(n)$
$\hat{X}(\omega_k)$	DFT de $x(n)$
r_{xx}	Autocorrelação do sinal limpo
i	Índice da iteração

$\tau(m)$	Suavização temporal
m	Índice do quadro
$\mathbf{y}$	Vetor do sinal ruidoso y
$\mathbf{x}$	Vetor do sinal de entrada x
$\mathbf{w}$	Vetor do sinal ruidoso w
F	Matriz de transformação DFT de M pontos
$\{\ .\ \}^H$	Operador Hermitiano
$\mathscr{H}$	Matriz $M \times M$ de *rank* completo
$\varepsilon(\omega)$	Vetor erro de estimação
$\varepsilon_x(\omega)$	Erro (distorção) referente a sinal limpo
$\varepsilon_w(\omega)$	Erro (distorção) referente ao ruído
I	Matriz identidade
δ	Limiar de distorção do ruído
μ	Multiplicador de Lagrange
L	Equação Lagrangeana
μ	Parâmetro que regula a atenução do filtro de Wiener
ξ_k	Relação sinal-ruído
$\hat{\xi}_k$	Estimativa da SNR
m	Índice do quadro
$p(m)$	Estimativa do *pitch*
N	Número de harmônicas de interesse
σ_g	Parâmetro que regula a abertura da concavidade
$A_{\max}$	Valor máximo de $\beta(\omega_k, m)$
$A_{\min}$	Valor mínimo de $\beta(\omega_k, m)$
$\beta(\omega_k, m)$	Parâmetro adaptativo do algoritmo proposto
ϕ	Fator de suavização
$A_{\max_o}$	Valor de $A_{\max}$ que maximiza os indicadores PESQ e CSII

SUMÁRIO

1 INTRODUÇÃO

Sinais de voz contaminados por ruído de fundo representam um dos maiores problemas em sistemas de comunicação. O ruído de fundo está presente constantemente ao nosso redor, como por exemplo, ruído de carros em ruas movimentadas, em restaurantes, shoppings, etc.

Em telefonia, esse tipo de interferência causa degradação do sinal de voz e com isso problemas de inteligibilidade, além de fadiga dos usuários. Para redução desses efeitos indesejáveis empregam-se algumas técnicas de redução de ruído. Estas técnicas já foram bastante exploradas e cada vez mais têm sido alvo de pesquisa (PLAPOUS; MARRO; SCALART, 2005). A área da engenharia que estuda essas técnicas é denominada melhoria da voz (do Inglês *speech enhancement*). Antes de explorar estas técnicas, é importante entender algumas características referentes ao ruído.

Os ruídos podem ser classificados basicamente como estacionários ou não estacionários. Na classe dos estacionários, podemos citar, por exemplo, o ruído proveniente dos *coolers* de computadores. A principal característica desse tipo de ruído é não sofrer variações em suas estatísticas ao longo do tempo. Já nos não-estacionários, as características temporais e espectrais mudam constantemente à medida que o ruído ambiente varia como, por exemplo, em restaurantes e ruas. Além da estacionaridade, outra característica importante dos vários tipos de ruído é a composição espectral, ou seja, a distribuição da energia do ruído no domínio da frequência. Diferentes tipos de ruído têm espectros de frequência diferentes. A maior parcela da energia pode estar concentrada em baixas frequências (barulho de vento), altas frequências (ventiladores) ou assumir uma distribuição mais uniforme no espectro (turbina de avião). O conhecimento *a priori* da distribuição da energia pode nos auxiliar na escolha do método de redução de ruído. Obviamente, os métodos existentes tem desempenhos diferenciados para cada aplicação. Geralmente

não sabemos de antemão qual a característica do ruído aditivo. Além disso, pode haver uma combinação de vários tipos de ruído. Portanto, o ideal é que os algoritmos funcionem da maneira mais abrangente possível.

Outra característica importante que deve ser levada em consideração na redução de ruído é a Relação Sinal-Ruído (SNR - Signal to Noise Ratio), que é a razão entre a potência do sinal de voz e a potência do ruído. A SNR varia de ambiente para ambiente. Quando é elevada, o sinal desejado é pouco afetado, ou seja, a potência do sinal é bem maior que a potência do ruído. Quando é baixa, o sinal desejado fica bastante prejudicado, pois a potência do ruído é muito próxima da potência do sinal ou até superior, prejudicando muito a compreensão. Tipicamente, considerando a potência média de um sinal de voz, em locais mais silenciosos a SNR é maior do que 30 dB. Para níveis inferiores a 20 *dB* (inclusive valores negativos) o ruído começa a prejudicar a inteligibilidade e, portanto, se faz necessário o uso de técnicas de redução de ruído. Nestes casos, os algoritmos de redução são indicados, pois reduzem o ruído sem cancelar o sinal de voz e, portanto, melhoram a SNR.

A literatura apresenta diversos métodos para equacionar o problema de cancelamento de ruído. Como exemplo podemos citar os filtros adaptativos, que podem ir dos mais simples aos mais complexos (SAID, 2008). As estruturas mais complexas podem usar mais de um microfone e aplicar técnicas chamadas de *beamforming*. O número de microfones pode influenciar bastante no desempenho dos algoritmos. Tipicamente, quanto maior o número de microfones melhores são os resultados.

Estruturas mais complexas e com mais microfones geralmente proporcionam melhores resultados, no entanto, esse desempenho tende a vir com um custo alto. Quanto mais microfones forem usados, maior a quantidade de informação disponível. Esse volume adicional de informação causa também o aumento do custo computacional, exigindo processadores com maior capacidade de processamento e memória. Esse custo computacional mais elevado

demanda hardwares mais complexos e mais caros, tornando às vezes a sua realização inviável.

Métodos convencionais de cancelamento de ruído empregados em sistemas de comunicação utilizam apenas um microfone. Esses métodos têm um custo computacional bem menor e portanto são bem mais baratos do que os métodos adaptativos com o uso de *beamforming*. A decisão de qual método usar é uma relação de compromisso entre custo e desempenho. Para sistemas que exijam processamento em tempo real, é interessante usar algoritmos mais rápidos, ou seja, métodos de baixa complexidade computacional. Este grupo de algoritmos possui muitas técnicas propostas ao longo dos anos e pode ser organizado em quatro classes básicas: subtração espectral (BOLL, 1979), subespaço (EPHRAIM; TREES, 1995), modelagem estatística (EPHRAIM; MALAH, 1984) e filtragem de Wiener (HU; LOIZOU, 2006). A última técnica é bastante popular devido a sua simplicidade e baixa complexidade computacional.

1.1 Classes básicas

1.1.1 Métodos baseados em modelos estatísticos

Esta classe de algoritmos dá ênfase a estimadores não-lineares de magnitude usando diversos modelos estatísticos e critérios de otimização. Estes estimadores não-lineares levam em conta a função de densidade de probabilidade (FDP) do ruído e dos coeficientes da DFT do sinal de voz. Várias são as técnicas existentes na literatura da teoria de estimação (KAY, 1993) para deduzir estes estimadores, que incluem estimadores de máxima verossimilhança e Bayesianos. Basicamente, eles diferem nas suposições feitas sobre os parâmetros de interesse, que podem ser modelados como determinísticos ou aleatórios.

1.1.2 Métodos baseados em subtração espectral

Historicamente, esta classe de algoritmos foi a mais explorada na literatura. Seu princípio de funcionamento é simples. Considerando um sinal de voz contaminado com ruído aditivo, pode-se obter uma estimativa do espectro do sinal de voz limpo subtraindo do espectro do sinal ruidoso uma estimativa do espectro do ruído. Comumente a estimativa do espectro do ruído é determinada através do espectro do sinal de voz ruidoso, utilizando um detector de atividade de voz (VAD). Durante os períodos de ausência de voz, uma estimativa do espectro do ruído é obtida e é atualizada constantemente à medida que novos períodos com ausência de voz são detectados. O sinal de voz melhorado é obtido através da transformada inversa de Fourier da subtração da estimativa do espectro do ruído e do espectro do sinal ruidoso.

1.1.3 Métodos baseados na projeção em subespaços

Esta classe explora fundamentos da álgebra linear para distinguir o que é sinal de voz do que é ruído. Os algoritmos pertencentes a essa classe se baseiam no princípio de que o espaço Euclidiano no qual o sinal ruidoso está contido, pode ser decomposto em dois subespaços complementares; um contendo o sinal de voz e outro contendo o ruído. Essa decomposição normalmente é feita utilizando duas técnicas principais: a decomposição em valores singulares (SVD) ou a decomposição em autovetores e autovalores (EVD). Uma técnica bastante usada baseia-se na matriz de autovetores da matriz de covariância do sinal, transformação comumente encontrada na literatura como Transformação de Karhunen-Loève ou simplesmente KLT. Este método simplesmente projeta o sinal ruidoso no subespaço do sinal sem fazer nenhuma outra modificação. Embora isto não acrescente distorções ao sinal, deixa uma quantidade razoável de ruído residual[1]. Para contornar esse

[1]Na prática, estimar a dimensão exata dos subespaços é um problema complexo. A imprecisão da dimensão pode permitir que o sinal de voz contenha mais ruído do que o desejado.

problema, o sinal projetado precisa ser modificado de forma a minimizar a energia residual do erro. Essa modificação é feita inserindo um filtro de Wiener, pois ele é capaz de minimizar a energia do erro. De forma genérica, podemos dizer que o que distingue os vários métodos, é o tipo de estimador de Wiener acrescentado para esta finalidade. Dentre os vários estimadores, podemos destacar: estimador de mínimos quadrados (LSE - Least Square Estimator) (EPHRAIM; TREES, 1995), estimador linear de mínimo erro quadrático médio (LMMSE - Linear Minimum Mean-Square Estimator), estimador com restrição no domínio tempo (TDCE - Time-Domain-Constrained Estimator) e estimador com restrição no domínio da frequência (SDCE - Spectral-Domain-Constrained Estimator). A implementação destes algoritmos requer alto custo computacional, pois tanto os valores singulares como os autovetores precisam ser calculados a cada quadro de sinal processado.

1.1.4 Filtragem de Wiener

A filtragem de Wiener é uma das técnicas mais importantes e usadas dentre as classes básicas devido à sua eficácia, robustez, baixa complexidade e fácil implementação. Apesar de melhorar a relação sinal-ruído, esse método tende a distorcer o sinal e ainda insere um fenômeno perceptual denominado ruído musical. Esta técnica apresenta bons resultados quando a relação sinal-ruído não é muito baixa. Porém, em baixas SNRs o ruído musical se torna mais perceptível. Além disso, problemas de distorção na voz também aparecem com mais frequência e intensidade, afetando a inteligibilidade.

Várias técnicas exploraram propriedades de mascaramento baseado em características do sistema auditivo, resultando em boa qualidade de sinal com níveis reduzidos de ruído musical. Como possuímos somente o sinal ruidoso, a potência do ruído deve ser estimada através do sinal ruidoso. Um dos motivos que causam o ruído musical é a obtenção de estimativas ruins da potência do ruído. Além disso, os algoritmos de redução de ruído geral-

mente não objetivam melhorar a inteligibilidade do sinal, que é a capacidade humana de compreender sons de fala (speech)(KJEMS et al., 2009). Em alguns casos, melhorar a qualidade pode vir acompanhada de uma diminuição da inteligibilidade, devido à distorção inserida do sinal de voz filtrado. Como esta técnica servirá de base para o trabalho proposto, faremos um análise mais detalhada desta técnica no Capítulo 2.

1.2 Objetivos

O presente trabalho propõe um método para atenuar alguns dos problemas existentes na filtragem de Wiener. Um dos maiores problemas encontrados em técnicas de redução de ruído é a inteligibilidade. Somado a isso, propomos a diminuição do ruído de fundo sem distorcer o sinal de voz e também atenuar o ruído musical.

1.3 Estrutura do trabalho

O Capítulo 2 apresenta a descrição teórica da filtragem de Wiener para cancelamento de ruído. Apresenta também uma variante do filtro de Wiener, chamada filtro de Wiener paramétrico. Além disso, problemas característicos do filtro e uma abordagem promissora de um artigo que servirá de ponto de partida para o presente trabalho. No Capítulo 3 está descrito o algoritmo proposto, funcionamento e parâmetros de projeto. O Capítulo 4 mostra uma avaliação estatística comparando os algoritmos convencional e proposto. Por fim, o Capítulo 5 apresenta as conclusões sobre o trabalho.

2 FILTRAGEM DE WIENER

Para explicar a filtragem de Wiener considere a Figura 1. O sinal de entrada $y(n)$ é o sinal com ruído, sendo equacionado de acordo com a Equação (2.1)

$$y(n) = x(n) + w(n) \tag{2.1}$$

em que n é o índice de tempo, $x(n)$ representa o sinal limpo, $w(n)$ representa o ruído aditivo indesejado e $y(n)$ é o sinal de entrada ruidoso. O objetivo do filtro de Wiener é produzir uma estimativa linear de mínimo erro quadrático do sinal limpo $x(n)$.

Estamos assumindo aqui que o sinal de entrada $y(n)$ e a resposta desejada $d(n)$ representam realizaçoes de processos estocásticos conjuntamente estacionários, onde o processo de estimação é acompanhado de um erro com características estatísticas. Podemos definir o erro de estimação $e(n)$ como:

$$e(n) = d(n) - \hat{d}(n) \tag{2.2}$$

em que $d(n)$ é a resposta desejada e $\hat{d}(n)$ é a estimativa da resposta desejada.

Para determinar a solução ótima de Wiener no domínio da frequência, a estimativa $\hat{d}(n)$ será modelada como a convolução entre o sinal de entrada

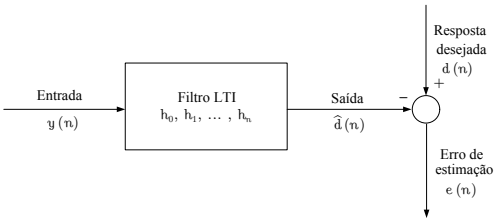


Figura 1: Diagrama de blocos do sistema

$y(n)$ e a resposta ao impulso de duração infinita $h(n)$ de um sistema linear

$$\hat{d}(n) = \sum_{k=-\infty}^{\infty} h_k \, y(n-k), \quad -\infty < \mathrm{n} < \infty \tag{2.3}$$

válida para todo n. Usando o símbolo $*$ para representar a convolução linear,

$$\hat{d}(n) = h(n) * y(n). \tag{2.4}$$

em que $h(n)$ representa um filtro LTI.

A transformada de Fourier de um processo estocástico $x(t)$ é um processo estocástico $X(\omega)$ dado por:

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \tag{2.5}$$

Esta integral é interpretada como um limite, no sentido médio quadrático e também vale para processos aleatórios (PAPOULIS, 1991). No domínio discreto podemos interpretar como

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \tag{2.6}$$

Portanto, aplicando a transformada de Fourier à equação (2.3) temos

$$\hat{D}(\omega) = H(\omega) Y(\omega) \tag{2.7}$$

onde $H(\omega)$, $Y(\omega)$ e $\hat{D}(\omega)$ são as transformadas de Fourier em tempo discreto (DTFT), respectivamente, de $h(n)$, $y(n)$ e $d(n)$.

Em aplicações práticas, o cálculo da DTFT é normalmente substituído pelo cálculo da DFT de M pontos, que proporciona uma representação espectral de dimensão finita na frequências $\omega_k = 2k\pi/M$ rad.

Transformando a Equação (2.2) e usando a Equação (2.7), para todo

$\omega = \omega_k = 2k\pi/M$, obtemos

$$\psi(\omega_k) = D(\omega_k) - H(\omega_k)Y(\omega_k) \tag{2.8}$$

em que $\psi(\omega_k)$ é a DFT de $e(n)$. Isto significa que teremos um erro de estimação para cada frequência ω_k.

Com o objetivo de obter $H(\omega)$ que minimiza o erro quadrático médio na frequência ω_k, devemos efetuar a minimização da seguinte maneira:

$$\frac{\partial(E\{|\psi(\omega_k)|^2\})}{\partial H(\omega_k)} = 0 \tag{2.9}$$

onde $E\{\cdot\}$ é o operador do valor esperado. Para encontrar soluções ótimas, devemos tirar o valor absoluto ao quadrado da Equação (2.8), calcular seu valor esperado, para posteriormente derivar e igualar a zero. Então, o erro quadrático médio pode ser expresso por

$$\begin{aligned} E\{|\psi(\omega_k)|^2\} &= E\{[D(\omega_k) - H(\omega_k)Y(\omega_k)]^*[D(\omega_k) - H(\omega_k)Y(\omega_k)]\} && (2.10)\\ &= E\{[|D(\omega_k)|^2] - H(\omega_k)E[D^*(\omega_k)Y(\omega_k)] && (2.11)\\ &\quad -H^*(\omega_k)E[Y^*(\omega_k)D(\omega_k)] + |H(\omega_k)|^2 E[|Y(\omega_k)|^2] && (2.12) \end{aligned}$$

onde a^* é o complexo conjugado de a. Derivando a Equação (2.12) e igualando a zero, obtemos:

$$[H(\omega_k)P_{yy}(\omega_k) - P_{dy}(\omega_k)]^* = 0 \tag{2.13}$$

onde $P_{yy}(\omega_k) = E\{|Y(\omega_k)|^2\}$ é o espectro de potência do sinal de entrada em que, $\omega = \omega_k$, $y(n)$ e $P_{yd}(\omega_k) = E\{Y(\omega_k)D^*(\omega_k)\}$ é o espectro de potência cruzado entre o sinal desejado $d(n)$ e o de entrada $y(n)$. Resolvendo para $H(\omega_k)$, obtemos a expressão para a resposta em frequência do filtro de Wiener

no domínio da frequência:

$$H(\omega_k) = \frac{P_{dy}(\omega_k)}{P_{yy}(\omega_k)}. \tag{2.14}$$

Em nossa aplicação, o sinal desejado $d(n)$ é o próprio sinal de voz $x(n)$. Portanto, temos que $D(\omega_k) = X(\omega_k)$ e podemos calcular $P_{dy}(\omega_k)$ usando a Equação (2.1) transformada para o domínio da frequência, ou seja,

$$P_{dy}(\omega_k) = E[X(\omega_k)\{X(\omega_k) + W(\omega_k)\}^*] \tag{2.15}$$

Teoricamente, $H(\omega)$ pode ser implementada usando um filtro de resposta infinita (IIR) ou um filtro de resposta finita (FIR). Em geral, o projeto utilizando filtros FIR é preferível pela maior simplicidade no controle de estabilidade e pela possibilidade de obter soluções com fase linear (HAYKIN, 2002). Para se obter os coeficientes deste filtro, utiliza-se o critério de minimização do erro quadrático médio .

Resolvendo o valor esperado de (2.15) resulta em:

$$P_{dy}(\omega_k) = P_{xx}(\omega_k) \tag{2.16}$$

onde foi assumido que o sinal e o ruído são descorrelacionados. O espectro de potência do sinal ruidoso $P_{yy}(\omega_k)$ pode ser calculado como

$$\begin{aligned} P_{yy}(\omega_k) &= E[\{X(\omega_k) + W(\omega_k)\}\{X(\omega_k) + W(\omega_k)\}^*] \\ &= P_{xx}(\omega_k) + P_{ww}(\omega_k) \end{aligned} \tag{2.17}$$

e, substituindo (2.16) e (2.17) em (2.14), resulta em:

$$H(\omega_k) = \frac{P_{xx}(\omega_k)}{P_{xx}(\omega_k) + P_{ww}(\omega_k)} \tag{2.18}$$

Definindo

$$\xi_k = \frac{P_{xx}(\omega_k)}{P_{ww}(\omega_k)} \tag{2.19}$$

como sendo SNR *a priori*, podemos expressar a Equação (2.18) como uma função da SNR *a priori*. Portanto, o filtro de Wiener se reduz a

$$H(\omega_k) = \frac{\xi_k}{\xi_k + 1}. \tag{2.20}$$

A partir da equação (2.20), podemos estimar o sinal filtrado $\hat{x}(n)$ aplicando a transformada de Fourier inversa à

$$\hat{X}(\omega_k) = H(\omega_k)Y(\omega_k). \tag{2.21}$$

É possível verificar que o ganho aplicado pelo filtro de Wiener é proporcional à SNR por componente de frequência, ou seja, o filtro dá ênfase à parte do espectro onde a SNR é alta e atenua quando a SNR é baixa.

A implementação prática dessa situação, no entanto, não é factível. Por definição, $P_{xx}(\omega_k)$ é a transformada de Fourier da autocorrelação do sinal limpo *x(n)*, r_{xx}, que não é conhecido *a priori*. Além do mais, o filtro de Wiener é não-causal e, portanto, não realizável em tempo real. A não-causalidade pode ser observada na Equação (2.18), pois tanto $P_{xx}(\omega_k)$ como $P_{ww}(\omega_k)$ são maiores ou iguais a zero e são duas funções pares. Portanto $H(\omega_k)$ é real e par, como consequência, a resposta ao impulso h_k será par também, e portanto, não causal.

Várias técnicas foram propostas para estimar o filtro de Wiener a partir do sinal ruidoso, como o método iterativo de Wiener. Este procedimento iterativo considera que o espectro melhorado do sinal $\hat{X}(\omega_k)$ é estimado através de

$$\hat{X}_{i+1}(\omega_k) = H_i(\omega_k)Y(\omega_k) \tag{2.22}$$

onde $H_i(\omega_k)$ representa o filtro de Wiener obtido na i-ésima iteração.

Este método iterativo, proposto por (LIM; OPPENHEIM, 1978), transforma o problema de redução de ruído em um problema de estimação de parâmetros auto-regressivos AR usados para gerar o sinal limpo. Esta técnica possui alguns problemas. Por exemplo, a definição do critério de convergência adotado e do ponto de parada do algoritmo, que é crucial para um bom desempenho. Para este último problema, é possível impor restrições de continuidade espectral, de forma a assegurar que o espectro obtido em um determinado quadro não seja muito diferente dos quadros anteriores ou posteriores. Estas restrições podem ser feitas de forma iterativa ou impostas na dedução do filtro de Wiener no sentido ótimo. Dentre os tipos de restrição mais comuns estão: *across-time* (QUATIERI; DUNN, 2002; QUATIERI; BAXTER, 1997) e *across-iterations* (HANSEN, 1988; HANSEN; CLEMENTS, 1991; SREENIVAS; KIRNAPURE, 1996).

Na restrição *across-time* busca-se aplicar uma suavização temporal,$\tau(m)$, no espectro estimado de $x(n)$. No m-ésimo quadro, $P_{xx}^m(\omega)$ é determinado por

$$P_{xx}^{m+1}(\omega) = \tau(m)P_{xx}^{m-1}(\omega) + (1-\tau(m))P_{xx}^m(\omega) \tag{2.23}$$

onde $\tau(m)$ é uma função de suavização que explora algumas caraterísticas do espectro. Quando o espectro varia rapidamente, podemos aplicar uma suavização temporal menor. Por outro lado, podemos aumentar a suavização quando o espectro é relativamente estacionário, por exemplo durante os quadros vozeados. A função $\tau(m)$ varia de acordo com uma medida de estacionaridade, que é estimada, e pode assumir valores no intervalo $0 \leq \tau(m) \leq 1$. As estimativas de $\tau(m)$ são baseadas no cálculo da diferença de primeira ordem entre o espectro atual e o passado.

As restrições *across-iterations* não são aplicadas diretamente aos coeficientes do modelo AR (Auto-Regressivo). São aplicadas nos coeficientes

de autocorrelação de $x(n)$, que são usados para deduzir os coeficientes do modelo AR de $x(n)$.

O procedimento iterativo provou melhorar a qualidade do sinal filtrado. Porém, sua aplicação em tempo real é inviável devido à necessidade de acesso a quadros futuros.

Ainda considerando filtros de Wiener iterativos, um novo método iterativo de redução de ruído utiliza um *codebook* de Densidade Espectral de Potência (DEP), formado utilizando periodogramas, de sinais limpos e diversos tipos de ruído (CHEHRESA; SAVOJI, 2009). Neste estudo, o algoritmo proposto estima as DEPs de fala e ruído e calcula a SNR resolvendo conjuntos de equações que possuem mais de uma solução. Uma variação do filtro de Wiener paramétrico também foi explorada, em que os parâmetros μ e β são calculados através de estatísticas de ordem superior, como *skewness* (assimetria) e *kurtosis* (curtose). O método iterativo com *codebook* teve resultados bons quando comparado ao método tradicional (WIENER, 1964). Já quando utilizado conjuntamente com o método de Wiener paramétrico, os resultados tiveram apenas uma pequena melhora. Apesar do controle dos parâmetros ser interessante, esses momentos de alta ordem são bastante custosos de se calcular. Além disso, a regra é aplicada igualmente para todo o espectro, ignorando a dependência espectral da SNR.

Uma alternativa para o método iterativo é incluir as restrições na dedução do filtro de Wiener ótimo no sentido quadrático médio usando critérios de restrição. Isto é feito minimizando a distorção do sinal[1] e restringindo a distorção do ruído abaixo de um limite. Seja $\mathbf{y} = \mathbf{x} + \mathbf{w}$ o sinal de voz ruidoso, sendo $\mathbf{x}$ o sinal limpo e $\mathbf{w}$ o ruído aditivo, onde as variáveis em negrito denotam vetores.

[1] As distorções do sinal de voz estão relacionadas à perda de informações relevantes do sinal de voz devido à atenuação demasiada empregada no processo de filtragem.

Definiremos F como a matriz de transformação DFT de M pontos

$$F = \frac{1}{\sqrt{M}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{j\omega_0} & \cdots & e^{j(M-1)\omega_0} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{j(M-1)\omega_0} & \cdots & e^{j(M-1)(M-1)\omega_0} \end{bmatrix}, \tag{2.24}$$

onde $\omega_0 = 2\pi/M$ (LOIZOU, 2007). Aplicando a matriz de transformação F ao sinal ruidoso $\mathbf{y}$ obteremos

$$\begin{aligned} \mathbf{Y}(\omega) &= F^H \mathbf{y} && (2.25) \\ &= F^H \mathbf{x} + F^H \mathbf{w} && (2.26) \\ &= \mathbf{X}(\omega) + \mathbf{W}(\omega) && (2.27) \end{aligned}$$

onde $\{ \, . \, \}^H$ indica o operador Hermitiano, $\mathbf{X}(\omega)$, $\mathbf{Y}(\omega)$ e $\mathbf{W}(\omega)$ possuem dimensão $M \times 1$ e contêm as componentes espectrais de $\mathbf{x}$, $\mathbf{y}$ e $\mathbf{w}$, respectivamente.

De forma geral, definiremos o estimador linear de $\mathbf{X}(\omega)$ como:

$$\hat{\mathbf{X}}(\omega) = \mathscr{H}\mathbf{Y}(\omega) \tag{2.28}$$

onde $\mathscr{H}$ é uma matriz $M \times M$ de *rank* completo. É interessante notar que a Equação (2.21) é um caso particular de (2.28) onde $\mathscr{H}$ é uma matriz diagonal com $\mathscr{H}_{kk} = H(\omega_k)$. Com isso, podemos definir o vetor erro de estimação $\varepsilon(\omega)$ com

$$\varepsilon(\omega) = \hat{\mathbf{X}}(\omega) - \mathbf{X}(\omega). \tag{2.29}$$

Então, substituindo a Equação (2.28) na Equação (2.29), podemos separar o erro de estimação em erro do sinal e erro do ruído

$$\begin{aligned}\varepsilon(\omega) &= \mathscr{H}\mathbf{Y}(\omega) - \mathbf{X}(\omega) \\ &= \mathscr{H}(\mathbf{X}(\omega) + \mathbf{W}(\omega)) - \mathbf{X}(\omega) \\ &= \underbrace{(\mathscr{H} - I)\mathbf{X}(\omega)}_{\varepsilon_x(\omega)} + \underbrace{\mathscr{H}\mathbf{W}(\omega)}_{\varepsilon_w(\omega)} \\ &= \varepsilon_x(\omega) + \varepsilon_w(\omega) \end{aligned} \tag{2.30}$$

onde $\varepsilon_x(\omega)$ é o erro (distorção) referente a sinal limpo, $\varepsilon_w(\omega)$ é o erro (distorção) referente ao ruído e I é a matriz identidade.

Assumindo $x(n)$ e $w(n)$ descorrelacionados e elevando o valor absoluto de cada um dos erros da Equação (2.30) ao quadrado e tirando o valor esperado, podemos obter a energia de distorção do sinal e a energia de distorção do ruído, apresentados respectivamente pelas Equações (2.31) e (2.32).

$$\varepsilon_x^2 = tr\{(\mathscr{H} - I)F^H R_{xx} F(\mathscr{H} - I)^H\} \tag{2.31}$$

$$\varepsilon_w^2 = tr\{\mathscr{H} F^H R_{ww} F \mathscr{H}^H\} \tag{2.32}$$

onde $R_{xx} = E\{X(\omega)X^H(\omega)\}$ e $R_{ww} = E\{W(\omega)W^H(\omega)\}$.

Conforme descrito anteriormente, o objetivo é minimizar a distorção do sinal mantendo a distorção do ruído abaixo de um limite δ. Isso é modelado como um problema de otimização com restrição conforme a Equação (2.33b)

$$\min_{\mathscr{H}} \varepsilon_x^2 \tag{2.33a}$$

$$\text{sujeito a: } \frac{1}{M}\varepsilon_w^2 \leq \delta \tag{2.33b}$$

podendo ser resolvido de diversas maneiras. Porém, o método dos multiplicadores de Lagrange é bastante apropriado para o caso, pois transforma um

problema com restrições em um problema sem restrições. Reorganizando (2.33b) e multiplicando por M, obtemos

$$\begin{aligned} &\min_{\mathscr{H}} \varepsilon_x^2 \\ &\text{sujeito a: } M\delta - \varepsilon_w^2 \geq 0. \end{aligned} \tag{2.34}$$

Portanto, a Equação (2.34) pode ser transformada em sua equação Lagrangeana (HIMMELBLAU, 1972) equivalente:

$$L(\mathscr{H}) = \varepsilon_x^2 - \mu(-\varepsilon_w^2 + M\delta) \tag{2.35}$$

onde μ é o multiplicador de Lagrange de L. Substituindo (2.31) e (2.32) em 2.35 obtemos:

$$\begin{aligned} L(\mathscr{H}) = \text{tr}[&\mathscr{H}F^H R_{xx} F\mathscr{H}^H - \mathscr{H}F^H R_{xx} FI - IF^H R_{xx} F\mathscr{H}^H + IF^H R_{xx} FI] + \\ &+ \mu(\text{tr}[\mathscr{H}F^H R_{ww} F\mathscr{H}^H] - M\delta) \end{aligned} \tag{2.36}$$

utilizando a propriedade de que o traço da soma é igual à soma dos traços teremos:

$$\begin{aligned} L(\mathscr{H}) = &\text{tr}[\mathscr{H}F^H R_{xx} F\mathscr{H}^H] - \text{tr}[\mathscr{H}F^H R_{xx} FI] - \text{tr}[IF^H R_{xx} F\mathscr{H}^H] + \text{tr}[IF^H R_{xx} FI] + \\ &+ \mu(\text{tr}[\mathscr{H}F^H R_{ww} F\mathscr{H}^H] - M\delta) \end{aligned} \tag{2.37}$$

Agora basta anularmos o gradiente de $L(\mathscr{H})$, ou seja,

$$\frac{\partial L(\mathscr{H})}{\partial \mathscr{H}} = 0. \tag{2.38}$$

Utilizando as propriedades de derivação em relação à matrizes (BERNSTEIN, 2005), obtemos

$$\frac{\partial L(\mathscr{H})}{\partial \mathscr{H}} = \mathscr{H}(F^H R_{xx} F)^H + \mathscr{H} F^H R_{xx} F - (F^H R_{xx} F)^H - F^H R_{xx} F + \tag{2.39}$$

$$+ \mu(\mathscr{H}(F^H R_{ww} F)^H + \mathscr{H} F^H R_{ww} F) = 0 \tag{2.40}$$

Assumindo que os processos $x(n)$ e $w(n)$ são estacionários e supondo R_{xx} e R_{ww} Hermitianos, podemos utilizar as propriedades:

$$(F^H R_{xx} F)^H = F^H R_{xx} F \tag{2.41}$$

$$(F^H R_{ww} F)^H = F^H R_{ww} F \tag{2.42}$$

para reorganizar a Equação 2.40, resultando em:

$$\mathscr{H}(F^H R_{xx} F + \mu F^H R_{ww} F) = F^H R_{xx} F \tag{2.43}$$

Assumindo que as matrizes R_{xx} e R_{ww} são Toeplitz e que o produto das matrizes das Equações (2.44a) e (2.44b) são assintoticamente (M $\to \infty$) diagonais para ,

$$\lim_{M\to\infty} [F^H R_{xx} F]_{ij} = 0, i \neq j \tag{2.44a}$$

$$\lim_{M\to\infty} [F^H R_{ww} F]_{ij} = 0, i \neq j \tag{2.44b}$$

então, denotando o k-ésimo elemento da diagonal principal de $\mathscr{H}$ como $[\mathscr{H}]_{kk}$, a Equação (2.43) pode ser simplificada conforme

$$[\mathscr{H}]_{kk} = \frac{P_{xx}(\omega_k)}{P_{xx}(\omega_k) + \mu P_{ww}(\omega_k)}. \tag{2.45}$$

A Equação (2.45) também pode ser expressa em termos da relação sinal ruído

ξ_k (Equação (2.19)), como:

$$[\mathscr{H}]_{kk} = \frac{\xi_k}{\xi_k + \mu} \tag{2.46}$$

onde o parâmetro μ desloca a curva atenuação para baixo à medida que μ aumenta. Como resultado, o parâmetro influencia a atenuação tanto em SNR alta como em SNR baixa.

Agora o problema fica restrito a se obter um boa estimativa de ξ_k, pois sabe-se que com uma boa estimativa de ξ_k (CAPPE, 1994) é possível eliminar o ruído musical, que por definição (BEROUTI; SCHWARTZ; MAKHOUL, 1979) é o ruído introduzido ao sinal pelo processo de retificação de meia onda do espectro de potência do sinal. Esse processamento não-linear dos pontos negativos gera picos isolados em frequências aleatórias que mudam a cada quadro e que, convertidos no tempo, soam como tons musicais de frequências aleatórias .

Para amenizar esse problema foram feitos vários estudos na literatura. Dentre as várias propostas existentes, uma das mais empregadas utiliza a estimativa da SNR *a priori* ξ_k dada pela combinação ponderada de estimativas presentes e passadas de ξ_k (método de *decisão-direcionada* (EPHRAIM; MALAH, 1984)), Essa estimativa é dada por

$$\hat{\xi}_k^m = \alpha \frac{\left|\hat{X}_k^{m-1}\right|^2}{\left|W_k^{m-1}\right|^2} + (1-\alpha)\max\left(\frac{\left|Y_k^m\right|^2}{\left|W_k^m\right|^2} - 1, 0\right) \tag{2.47}$$

onde $\hat{\xi}_k$ é a estimativa da SNR e W é substituído pela estimativa do ruído. Existem vários métodos para obter a estimativa do ruído, entre elas, através de um detector de atividade de voz.

2.1 Filtro de Wiener paramétrico

Podemos escrever o filtro de Wiener de uma forma mais genérica, chamada de filtro de Wiener paramétrico, conforme a equação abaixo:

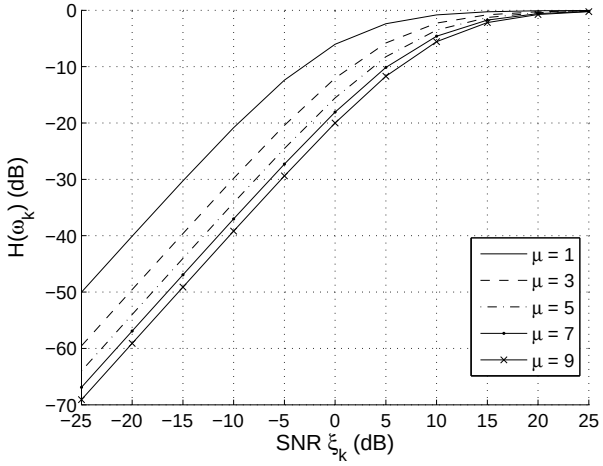
$$\hat{X}(\omega_k) = \left(\frac{P_{xx}(\omega_k)}{P_{xx}(\omega_k) + \mu P_{ww}(\omega_k)} \right)^{\beta} Y(\omega_k) \tag{2.48}$$

onde β e μ são escalares. O parâmetro β foi deduzido a partir da Equação (2.45) com o intuito de prover mais flexibilidade ao filtro. Variando de maneira adequada os parâmetros μ e β podemos obter filtros com comportamentos diferentes do tradicional. Essa generalização foi bastante explorada e está bem fundamentada em (LIM; OPPENHEIM, 1979).

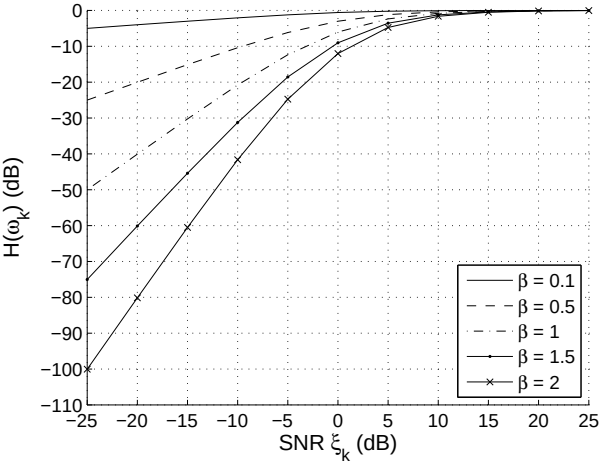
De maneira geral, podemos utilizar quaisquer valores para β e μ para regular a atenuação do filtro. Mantendo $\beta = 1$, podemos notar (ver Figura 2(a)) que aumentando μ, as curvas de atenuação são deslocadas para baixo e têm influência tanto em SNR alta como em baixa. Por outro lado, mantendo $\mu = 1$, podemos notar que quando maior o valor de β, maior a atenuação para relações sinal-ruído baixas, enquanto que para SNR a partir de 0 dB, a atenuação é baixa, tendendo a zero à medida que β cresce, conforme pode ser visto na Figura 2(b). Observando as curvas, concluímos que alterando os parâmetros β e μ podemos variar a atenuação do filtro. Quanto maior o valor dos parâmetros, maior a atenuação e, consequentemente, podemos diminuir o ruído residual no sinal. Porém, ao diminuirmos o ruído, aumentaremos também a distorção do sinal, já que a atenuação é aplicada para todo ω_k.

2.1.1 Raiz quadrada do filtro de Wiener

Para fazer uma análise do espectro de potência de $\hat{X}(\omega_k)$, utilizaremos a Equação (2.48) com o valor de $\mu = 1$, pois é o valor tradicionalmente utilizado. Portanto temos



(a) Curvas de atenuação $\beta = 1$



(b) Curvas de atenuação $\mu = 1$

Figura 2: Curvas de atenuação

$$\hat{X}(\omega_k) = \left(\frac{P_{xx}(\omega_k)}{P_{xx}(\omega_k)+P_{ww}(\omega_k)}\right)^{\beta} Y(\omega_k) \quad (2.49)$$

$$\hat{X}(\omega_k) = (H(\omega_k))^{\beta} Y(\omega_k) \quad (2.50)$$

Para $\beta = 0,5$ temos a raiz quadrada do ganho $H(\omega_k)$ filtro. Elevando (2.50) ao quadrado dos dois lados e tirando o valor esperado, temos

$$\begin{aligned} E\{|\hat{X}(\omega_k)|^2\} &= (\sqrt{H(\omega_k)})^2 E\{|Y(\omega_k)|^2\} \\ P_{\hat{x}\hat{x}}(\omega_k) &= H(\omega_k) P_{\hat{y}\hat{y}}(\omega_k) \\ P_{\hat{x}\hat{x}}(\omega_k) &= \frac{P_{xx}(\omega_k)}{P_{xx}(\omega_k)+P_{ww}(\omega_k)} P_{yy}(\omega_k). \end{aligned} \quad (2.51)$$

Como estamos supondo que o sinal e o ruído são descorrelacionados, podemos substituir $P_{yy}(\omega_k) = P_{xx}(\omega_k) + P_{ww}(\omega_k)$ em (2.51). Então,

$$\begin{aligned} P_{\hat{x}\hat{x}}(\omega_k) &= \frac{P_{xx}}{P_{xx}(\omega_k)+P_{ww}(\omega_k)} (P_{xx}(\omega_k)+P_{ww}(\omega_k)) \\ P_{\hat{x}\hat{x}}(\omega_k) &= P_{xx}(\omega_k). \end{aligned} \quad (2.52)$$

Isso significa que, para um sinal contaminado com ruído, o espectro de potência do sinal estimado $\hat{X}(k)$ é idêntico ao sinal limpo $X(k)$ quando se usa $\beta = 0,5$ (LOIZOU, 2007). Podemos ver que a raiz quadrada do filtro de Wiener é um caso particular da Equação (2.48) quando $\mu = 1$ e $\beta = 0,5$. Esse valor é particularmente importante para o algoritmo proposto, o qual será discutido posteriormente.

2.2 Problemas do filtro de Wiener

Técnicas baseadas na filtragem de Wiener que empregam funções de ganho exponenciais invariantes no tempo e na frequência podem gerar distorções e ruído musical no sinal filtrado resultante, sendo ambos desagradáveis.

É por isso que os estudos mais recentes têm favorecido o uso de funções de ganho mais flexíveis (AMEHRAYE; PASTOR; TAMTAOUI, 2008; CHEN; LOIZOU, 2010; PLAPOUS; MARRO; SCALART, 2005; DING et al., 2009).

O ruído musical é causado por picos espalhados no espectro de forma aleatória em frequências que diferem de quadro para quadro. Esses picos são gerados por estimativas erradas do espectro do sinal de ruído (HU; LOIZOU, 2004b). Quando convertidos para o domínio do tempo, esses picos tornam-se senoides de frequências aleatórias que soam como tons que são ligados e desligados.

Diversas técnicas foram propostas para eliminar o ruído musical, dentre elas está o bem conhecido método de *decisão-direcionada* (SCALART; FILHO, 1996) que usa uma suavização da estimativa da SNR *a priori* através de uma ponderação da estimativa do quadro atual com a estimativa do quadro anterior. Há também o método do limite inferior para a SNR *a priori* (CAPPE, 1994) e também um pós-filtro (ESCH; VARY, 2009) que suaviza os ganhos em frequência baseado em *decisões-suaves*. Outro método que utiliza pós-filtro combinado com o filtro de Wiener explora um método de pós-processamento com um limiar modificado de mascaramento para reduzir o ruído musical gerado pelo filtro de Wiener (e outros métodos clássicos) em cada banda crítica. Resultados experimentais comprovam a eficácia do método proposto (ALAM; O'SHAUGHNESSY, 2010) associado ao método de decisão-direcionada para estimar a SNR *a priori*. Há também um estimador linear (HU; LOIZOU, 2004a), que incorpora propriedades de mascaramento para o sistema de audição humana, tornando o ruído musical inaudível. Outro método usa propriedades de mascaramento para melhorar a percepção auditiva. O estudo (LU, 2007) adota um fator de ganho perceptual que considera propriedades de mascaramento do ruído *intra-frame* para suprimir o ruído de fundo. Adicionalmente, um fator de suavização *intra-frame* é deduzido para suavizar o espectro em quadros subsequentes para os quadros dominados por ruído e sub-bandas de

baixa energia. Isto faz com que o espectro suavizado varie uniformemente nos quadros dominados por ruído e regiões de baixa energia, consequentemente amenizando o fenômeno do ruído musical.

A inteligibilidade é outro problema bastante comum e de difícil resolução. Em busca das razões pelas quais os algoritmos de redução de ruído atuais não melhoram a inteligibilidade, Loizou (LOIZOU; KIM, 2010) impôs uma restrição no espectro de potência para analisar tipos de distorção existentes quando se trata de redução de ruído. As distorções no sinal de voz foram então classificadas em três regiões utilizando uma versão para o domínio da frequência da bem conhecida medida *segmental SNR*. Baseado em estudos feitos (MA; HU; LOIZOU, 2009) esta medida foi utilizada devido a sua alta correlação ($r = 0,81$) com a inteligibilidade e é definida como:

$$SNR_{ESI}(\omega_k) = \frac{X^2(\omega_k)}{(X(\omega_k) - \hat{X}(\omega_k))^2} \tag{2.53}$$

dividindo o numerador e o denominador por $(W(\omega_k))^2$ temos

$$SNR_{ESI}(\omega_k) = \frac{SNR(\omega_k)}{\sqrt{SNR(\omega_k)} - \sqrt{SNR_{ENH}(\omega_k)}}$$

onde SNR_{ENH} é a SNR do sinal filtrado[2]. Então, definindo $\varepsilon(k) = X(k) - \hat{X}(k)$ notamos que

$$\begin{cases} \text{se } \hat{X}(\omega_k) < X(\omega_k) & \varepsilon(k) > 0 \\ \text{se } \hat{X}(\omega_k) > X(\omega_k) & \varepsilon(k) < 0 \end{cases}$$

Em estudos feitos por Loizou (LOIZOU; KIM, 2010), a distorção da voz foi dividida em três regiões, especificadas abaixo:

- **Região I** - Compreende os valores onde $\hat{X}(\omega_k) \leq X(\omega_k)$, que sugere distorção por atenuação;

[2]Note que SNR_{ENH} não é a mesma coisa que SNR do sinal de saída, pois o ruído não é processado separadamente pelo algoritmo de redução de ruído

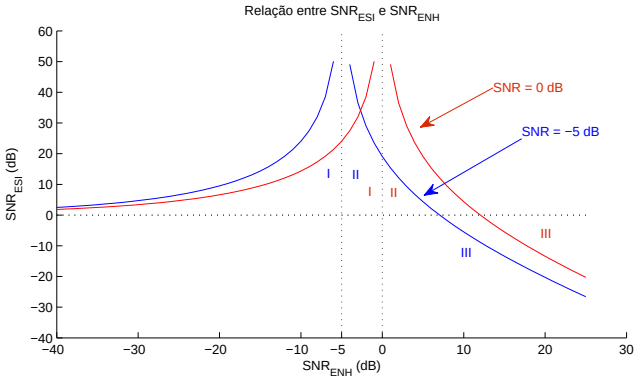


Figura 3: Identicação das regiões I, II e III

- **Região II** - Compreende os valores onde $X(k) < \hat{X}(\omega_k) \leq 2X(\omega_k)$, que sugere distorção por amplificação até $6,02\,dB$;
- **Região III** - Compreende os valores onde $\hat{X}(\omega_k) > 2X(\omega_k)$, que sugere distorção por amplificação maior que $6,02\,dB$.

Graficamente, podemos identificar essas regiões conforme a Figura 3.

As Regiões I e II (distorção por atenuação) se destacaram por ter pequeno impacto na inteligibilidade. A Região III (distorção por amplificação), entretanto, se mostrou a mais crítica por corresponder a um nível maior de distorção do sinal de voz. A Região III corresponde a estimativas do espectro do sinal de voz $\hat{X}(\omega_k)$ tal que $\hat{X}(\omega_k) > 2X(\omega_k)$, onde $X(\omega_k)$ é o espectro do sinal de voz limpo. A restrição a seguir foi então sugerida em (LOIZOU; KIM, 2010):

$$\text{se } \hat{X}(\omega_k) > 2X(\omega_k) \Rightarrow \hat{X}(\omega_k) = 0 \tag{2.54}$$

Reproduzindo as descobertas em seu estudo (LOIZOU; KIM, 2010), podemos plotar os espectros de potência do sinal limpo, sinal filtrado com o filtro de Wiener tradicional e também o mesmo sinal filtrado com a inclusão

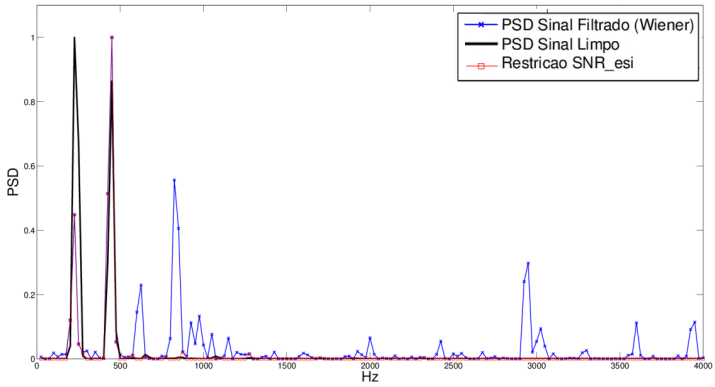


Figura 4: Espectro de Potência de Sinais Filtrados

da restrição (2.54). Com todas as curvas sobrepostas, podemos analisar, em frequência, o que desaparece no espectro do sinal processado quando incluímos a restrição. Para nos auxiliar ainda mais, o espectro de potência do sinal limpo nos servirá como referência de análise.

Podemos ver na Figura 4 que entre 200 a 1500 Hz, tanto o sinal filtrado com o filtro Wiener tradicional, quando utilizando a restrição (2.54) levam a espectros sobrepostos. Além disso, estes espectros estão muito próximos do espectro do sinal real limpo. A partir de 1500 Hz notamos picos que não fazem parte do sinal limpo. Esses picos além de gerarem distorções, também são responsáveis pelo ruído musical. Analisando a curva correspondente ao sinal ao qual foi aplicada a restrição (2.54), notamos que essas componentes foram zeradas e, consequentemente, a qualidade do sinal resultante será muito superior em termos de inteligibilidade.

Uma característica que podemos extrair facilmente dos sinas de voz é o *pitch*. *Pitch* é um fenômeno perceptual que permite ordenar os sons em uma escala musical. Entretanto, nem todos os sons possuem *pitch*. Quando falamos ou cantamos, alguns sons produzem uma sensação forte de *pitch*, por

exemplo as vogais. Outros, porém, não possuem (CAMACHO, 2008).

Com essa característica podemos determinar a frequência fundamental do sinal e, com isso, determinar a região de maior energia do sinal. Desta forma, podemos variar convenientemente a atenuação em cada ω_k para cada quadro como uma função do valor do *pitch*.

Apesar de o critério proposto (LOIZOU; KIM, 2010) ter levado a uma excelente melhora em inteligibilidade, o espectro limpo verdadeiro $X(k)$ não está disponível em situações práticas. Mesmo assim, esses resultados estabelecem um objetivo claro a ser seguido para obtermos algoritmos implementáveis na prática com desempenho semelhante.

No próximo capítulo, utilizaremos os conhecimentos do filtro de Wiener paramétrico como base para uma melhoria do filtro de Wiener , pois os parâmetros μ e β tem influência direta na atenuação empregada no sinal ruidoso. Se esses parâmetros forem modificados corretamente e de maneira controlada, podemos garantir uma qualidade superior ao sinal de voz resultante da filtragem.

3 MÉTODO PROPOSTO

Conforme apresentado no Capítulo 2, o filtro de Wiener paramétrico (LOIZOU, 2007) fornece uma função de ganho na frequência ω_k para o m-ésimo quadro de fala dado por

$$H(\omega_k,m)^\beta = \frac{\hat{X}(\omega_k,m)}{Y(\omega_k,m)} = \left(\frac{\xi(\omega_k,m)}{1+\xi(\omega_k,m)}\right)^\beta \tag{3.1}$$

onde $\hat{X}(\omega_k,m)$ é o espectro de potência estimado para o m-ésimo quadro na frequência ω_k, $Y(\omega_k,m)$ é o espectro de potência do sinal ruidoso e $\xi(\omega_k,m)$ é a relação sinal ruído *a priori*, que pode ser determinada através do método de decisão-direcionada (EPHRAIM; MALAH, 1984). O parâmetro β é fixo para todos os quadros e para toda a banda de frequência, sendo o valor típico igual a 0,5 (ver Secão 2.1). O parâmetro μ assumiu o valor unitário para que a propriedade (2.52) tenha validade.

Normalmente, o ruído residual pode ser reduzido com o aumento do valor de β. No entanto, isso introduz distorções indesejadas ao sinal, pois a atenuação extra é igualmente aplicada para todo ω_k e para todo m.

Para prover mais flexibilidade à função de ganhos, propomos utilizar em (3.1) um parâmetro $\beta(\omega_k,m)$, função agora de ω_k e m. Como a maior parte da energia do sinal de voz está concentrada em torno do *pitch* e de suas primeiras harmônicas, propomos ajustar o valor de β dentro de cada quadro de acordo com o valor do *pitch* estimado para cada quadro. Como o valor do *pitch* pode ser extraído facilmente do sinal de voz (CAMACHO, 2008), sua utilização na determinação de $\beta(\omega_k,m)$ não deve aumentar significativamente a complexidade computacional em relação ao filtro de Wiener convencional.

3.1 Estimando o *Pitch*

Há vários algoritmos para estimar o *pitch* de um sinal de voz. Em nosso estudo utilizamos um algoritmo que calcula o *Subharmonic-to Harmonic Ratio* (SHR) (SUN, 2002) após deslocamento espectral em escala de frequência logarítmica. O SHR é robusto ao ruído mesmo em SNRs muito baixas. De qualquer modo, o algoritmo proposto neste trabalho não é muito sensível a erros na detecção exata do *pitch*, pois necessitamos apenas uma estimativa das regiões do espectro que contêm a maior parte da energia do sinal de voz.

3.2 O parâmetro $\beta(\omega_k, m)$ adaptativo

Dada a estimativa de *pitch* para cada quadro, propomos utilizar a função $\beta(\omega_k, m)$ de forma a enfatizar as contribuições espectrais do entorno da frequência de *pitch* e de suas $N-1$ primeiras harmônicas. A atenuação espectral, então, aumenta progressivamente para as frequências distantes dessas harmônicas. Uma função com as características desejadas é

$$\beta(\omega_k, m) = f(\omega_k, m)(A_{\max} - A_{\min}) + A_{\min} \tag{3.2a}$$

com

$$f(\omega_k, m) = \prod_{n_h=1}^{N} \left\{ 1 - \exp\left[\frac{-[\omega_k - n_h p(m)]^2}{2\sigma_g^2} \right] \right\} \tag{3.2b}$$

em que m indexa o quadro, $p(m)$ é a estimativa do *pitch* para cada quadro m (SUN, 2002), N é o número de harmônicas de interesse e σ_g é o parâmetro que regula a abertura da concavidade. Cada termo da Equação 3.2 tem a forma de uma Gaussiana invertida. Esta forma foi escolhida por sua suavidade e conveniência matemática, pois pode ser facilmente centrada em cada estimativa do *pitch* $p(m)$ com a abertura da concavidade controlada por σ_g^2.

$A_{\max}$ e $A_{\min}$ são os valores máximo máximo e mínimo, respectivamente, do parâmetro $\beta(\omega_k, m)$.

A função $f(\omega_k, m)$ terá zeros nas harmônicas (até ordem N) do *pitch*. Em torno de cada frequência $n_h p(m)$ o valor de $f(\omega_k, m)$ cresce de acordo com um pulso gaussiano. Essa forma de pulso foi escolhida pela facilidade de ajuste de localização e de largura do pulso. Os valores de $A_{\max}$ e $A_{\min}$ completam a definição do parâmetro $\beta(\omega_k, m)$ de forma que $\min\{\beta(\omega_k, m)\} = A_{min}$ e $\max\{\beta(\omega_k, m)\} = A_{\max}$. Como $|H(\omega_k, m)|$ em (3.1) é menor do que 1, $\beta(\omega_k, m) = A_{\min}$ corresponderá à mínima atenuação de $H(\omega_k)$ (aplicada às frequências múltiplas do *pitch*). Já $\beta(\omega_k, m) = A_{\max}$ levará à máxima atenuação de $H(\omega_k, m)$ (aplicada às frequências distantes das harmônicas do *pitch*.

Como exemplo, a Figura 5 mostra a função $\beta(\omega_k, m)$ (curva de cor preta na parte superior do gráfico) para dois quadros de sinal de voz capturados aleatoriamente e para $N = 3$, $A_{\max} = 1$ e $A_{\min} = 0,5$. Esses valores de $A_{\max}$ e $A_{\min}$ foram utilizados para o exemplo pois foi verificado experimentalmente que atenuações no intervalo [0,5, ; 1] proporcionam bons resultados práticos. Note que as principais componentes espectrais são preservadas, enquanto que as componentes de frequências mais altas são mais atenuadas. Esta regra foi aplicada igualmente para quadros vozeados e não-vozeados. Em quadros não-vozeados não faz sentido pensar em *pitch*. Neste caso a estimativa de *pitch* é geralmente bem baixa (próxima de zero), levando a $\beta = A_{\max}$ para a maior parte do espectro. Se um discriminador para quadros vozeados/não-vozeados for empregado, pode-se aplicar $\beta = A_{\min}$ para quadros não-vozeados e, desta forma, melhorar a performance geral. Para períodos de silêncio uma nova regra é aplicada.

Para cada quadro de silêncio detectado pelo VAD, forçamos $A_{\min} =$

$A_{\max}$. Então, $A_{\min}$ é aumentado segundo a expressão:

$$A_{\min}(m,k) = \phi A_{\min}(m,k-1) + (1-\phi)A_{\min}(m-1,K) \tag{3.3}$$

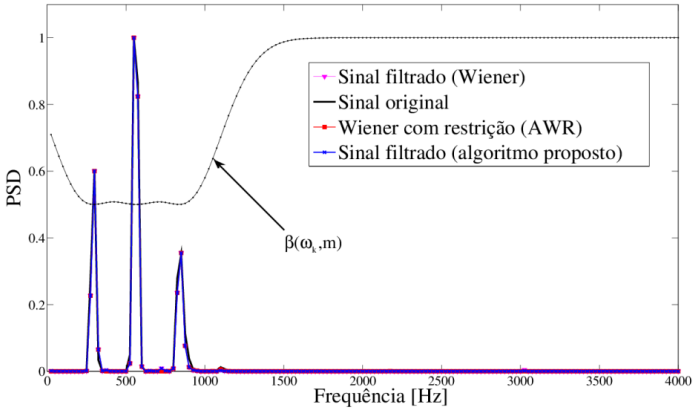
em que $0 \leq \phi \leq 1$ é o fator de suavização, k é o índice da frequência ω_k e K é o número de frequências por quadro. Testes feitos mostram que o melhor valor para este fator é $\phi = 0,3$. Quando o VAD indica presença de voz novamente, $A_{\min}$ é reinicializado com o valor inicial, por exemplo $A_{\min} = 0,5$ e reduzido progressivamente (até retornar ao valor mínimo) utilizando a Equação 3.3. Isto torna a transição mais suave entre os quadros com e sem presença de voz.

3.2.1 Propriedades do sinal de voz

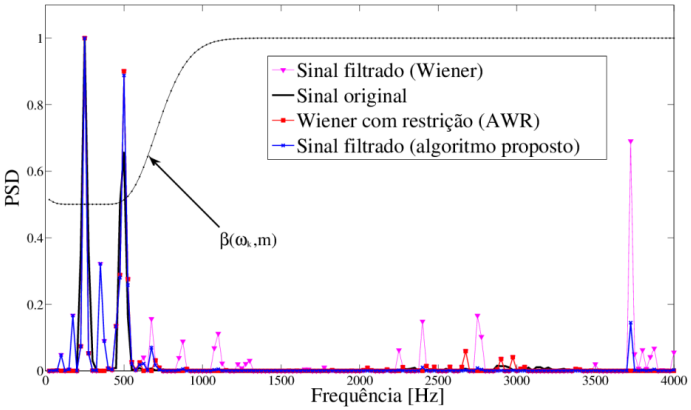
Como sabemos, os sinais de voz podem ser vozeados e não-vozeados. Sons vozeados são produzidos pela vibração das cordas vocais. Essa vibração ocorre quando o ar proveniente dos pulmões passa por uma abertura entre as duas cordas, chamada de glote. Por definição (QUATIERI, 2002), o tempo de duração de um ciclo glotal é chamado de período de *pitch* e o recíproco é chamado simplesmente de pitch[1] ou frequência fundamental. Existem três estados primários das cordas vocais: respiração, vozeado e não-vozeado. Na respiração, o ar proveniente dos pulmões passa livremente pela glote e pelas cordas vocais, portanto não há formação de som. Sons vozeados (como /u/ em az*u*l, por exemplo[2]) são produzidos por uma excitação quase periódica no trato vocal, por isso existe uma frequência fundamental (*pitch*). Nos sons não-vozeados (como /f/ em frequência, por exemplo) são gerados por um fluxo de ar turbulento através de uma constrição no trato vocal e, portanto, não há uma frequência fundamental associada.

[1] O *pitch* é definido pela ANSI (American National Standard Institute) como um atributo da sensação auditiva ligado à frequência do estímulo sonoro (PLACK, 2005)

[2] Aqui o o símbolo / . / indica um fonema, unidade básica da linguística.



(a) Quadro m_1.



(b) Quadro m_2.

Figura 5: Espectro de potência dos quadros m_1 e m_2 aleatoriamente selecionados.

3.3 Medidas objetivas

Testar a qualidade de sinais processados com algoritmos de filtragem para redução de ruído torna-se uma tarefa exaustiva se usarmos nossa audição para verificar possíveis melhoras. Além disso, é comum que pessoas tenham diferentes pontos de vista em relação ao que escutam, tornando esse método bem impreciso. Visando padronizar essa tarefa, surgiram as medidas objetivas, que procuram medir a qualidade e a inteligibilidade dos sinais de voz. Algumas das medidas objetivas são mais adequadas para medir qualidade, outras, são mais indicadas para verificar a inteligibilidade. Esta seção apresenta as medidas mais utilizadas e adequadas a cada necessidade.

3.3.1 Medida objetiva para qualidade

Dentre as medidas objetivas existentes, o PESQ é recomendado pela Recomendação ITU-T P.862 (ITU-T, 2001) para avaliar a qualidade de voz em aplicações de telefonia e é uma das mais complexas para calcular. Poucos estudos (BEERENDS et al., 2004; BEERENDS; WIJNGAARDEN; BUUREN, 2005) testaram essa medida para avaliar inteligibilidade (MA; HU; LOIZOU, 2009), porém, uma alta correlação ($r \approx 0,79$) foi obtida para julgamentos de qualidade subjetiva e de inteligibilidade para sinais processados por algoritmos de redução de ruído (HU; LOIZOU, 2008). O PESQ produz escores entre 1,0 e 4,5, sendo que os valores altos indicam melhor qualidade.

3.3.2 Medida objetiva para inteligibilidade

Como explicado anteriormente, o PESQ pode servir também para a avaliação de inteligibilidade. No entanto, sua correlação com a percepção subjetiva pode não ser elevada. Além desta medida, há também outras especificamente designadas para esse propósito. Dentre todas as medidas consideradas (MA; HU; LOIZOU, 2009), a que obteve melhor comportamento foi a

media CSII baseada em coerência, devido a sua alta correlação com a inteligibilidade. Estudos anteriores (KATES; AREHART, 2005; AREHART et al., 2007) também indicam que há alta correlação tanto em qualidade de voz quanto em inteligibilidade. Portanto, usaremos o CSII juntamento com o PESQ na avaliação da inteligibilidade. O CSII produz escores entre 0 e 1, sendo que valores mais altos indicam maior inteligibilidade.

3.4 Pré-processamento dos sinais de voz

Os algoritmos foram avaliados utilizando dois conjuntos de sinais de voz. O Conjunto de Treinamento (CT), usado para determinar os parâmetros de projeto, é composto por um total de 28 sinais de voz, sendo 14 sinais de voz masculina e 14 sinais de voz feminina. O Conjunto de Validação (SANTOS; ALCAIM, 1997) (CV), usado para avaliar desempenho estatístico, é composto por 24 sinais de voz masculina e 24 sinais de voz feminina. Cada sinal foi gravado a uma taxa de 22 kHz e depois reamostrado para 8 kHz. O sinais foram gravados por dois locutores diferentes na mesma proporção, como recomendado pela ITU-T P.830 (ITU-T, 1996), em língua Portuguesa, com duração média de 8 segundos e compostos por sentenças foneticamente balanceadas. Tanto os sinais quanto os ruídos foram filtrados utilizando o *filtro IRS modificado*, descrito na ITU-T P.862 (ITU-T, 2001), para simular a resposta em frequência dos dispositivos de telefonia. O filtro foi aplicado a cada um dos sinais de voz e a cada um dos sinais de ruído de forma independente. O sinal de voz limpo foi normalizado utilizando o Nível de Fala Ativa (*Active Speech Level* em Inglês) de acordo com o método B da ITU-T P.56 (ITU-T, 1993), para atingir a especificação de $-27 \pm 1\,\text{dBov}$[3]. Mais detalhes sobre esse procedimento podem ser encontrados no Anexo B.

O Conjunto de Ruídos (SILVA, 2010) (CR) é composto por ruído de

[3] A medida dBov é definida pela ITU-t como sendo a medida sonora máxima em decibéis com respeito a uma palavra de 16 bits, ou seja, é a máxima representação referente ao número de bits por palavra.

estação de trem, ruído de ventilador, ruído de restaurante e ruído de aspirador de pó, que foram gravados previamente em situações reais. Os sinais de ruído foram adicionados ao sinais limpos com potências apropriadas para produzir SNRs variando de $+20$ dB a -10 dB.

3.5 Escolha dos parâmetros de controle

Esta seção discute a escolha dos parâmetros N, σ_g, $A_{\max}$ e $A_{\min}$ que determinam o comportamento da função $\beta(\omega_k, m)$. O valor de $A_{\min}$ utilizado na Equação 3.2a será empegado no *pitch* e suas $N-1$ primeiras harmônicas. Portanto, deve receber um valor tal que não insira distorções ao sinal de voz. Como visto na Seção 2.1.1, $A_{\min} = 0,5$ possui características importantes em frequência e, portanto, será atribuído esse valor para todos os quadros. A escolha dos demais parâmetros é baseada em medidas objetivas para qualidade de voz e inteligibilidade. Nós usamos PESQ (ITU-T P.862 para telefonia) para avaliar qualidade da voz. Para avaliar a inteligibilidade, nós usamos o CSII (MA; HU; LOIZOU, 2009) devido à sua alta correlação com resultados subjetivos. A fim de determinar o conjunto ótimo de parâmetros, definimos $A_{\max_o}$ como sendo o valor de $A_{\max}$ para o qual o PESQ e CSII atingem seus maiores valores.

3.5.1 Influência de σ_g em $A_{\max_o}$

A Figura 6 mostra um exemplo da variação do valor do PESQ como uma função de $A_{\max}$ para $\sigma_g = 125$ e $\sigma_g = 245$, $N = 3$ e para diferentes relações sinal-ruído (SNR). Os valores de σ_g e N foram escolhidos baseados nos resultados de testes experimentais. Os resultados mostrados foram obtidos utilizando somente as sentenças de voz feminina do conjunto CT. Essas curvas indicam que, para o PESQ, $A_{\max_o}$ fica no intervalo $(1,5; 2,0)$. Resultados semelhantes foram obtidos para sinais de voz masculina e também considerando os dois casos juntos, para uma ampla faixa de relações sinal-

ruído. Então, é de se esperar que a escolha de $A_{\max}$ dentro desse intervalo seja robusto a mudanças no tipo de voz (masculina ou feminina) ou na SNR. Essa propriedade leva à redução de um grau de liberdade no projeto quando utilizamos valores de $A_{\max}$ dentro do intervalo determinado.

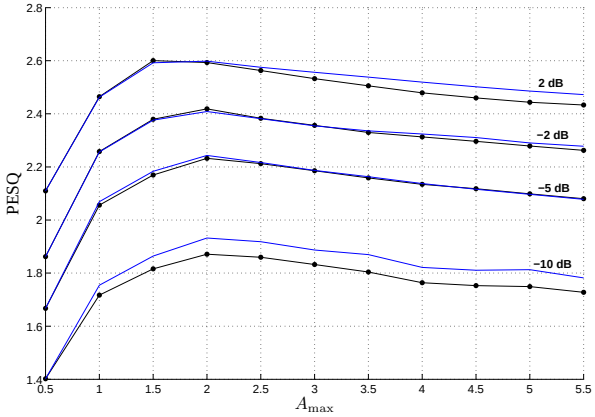


Figura 6: Influência de σ_g em $A_{\max_o}$ utilizando o indicador PESQ. Curvas com marcadores: $\sigma_g = 125$. Curvas sem marcadores: $\sigma_g = 245$. Sinal de voz feminina.

3.5.2 Influência do número de harmônicas usados para $\beta(\omega_k, m)$

A Figura 7 mostra o PESQ como uma função de $A_{\max}$ para $\sigma_g = 245$, $N = 3$ e $N = 5$, e diferentes relações sinal-ruído. Novamente, esses valores foram escolhidos experimentalmente para um bom desempenho. Esses resultados foram obtidos utilizando segmentos de voz feminina, mas resultados semelhantes foram obtidos para ambos os sexos. Note que o intervalo de valores de $A_{\max}$ para o qual o PESQ é maximizado é basicamente o mesmo verificado na Figura 6, independentemente da SNR.

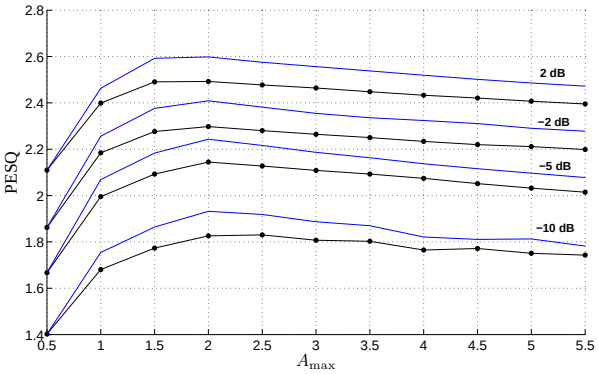


Figura 7: Influência de N em $A_{\max_o}$ utilizando o indicador PESQ. Curvas com marcadores: $N = 5$. Curvas sem marcadores: $N = 3$. Sinal de voz feminina.

3.6 Superfícies de desempenho

As superfícies de desempenho mostram os valores do PESQ e do CSII em função dos parâmetros $A_{\max}$ e N. Essas curvas foram geradas utilizando o conjunto CT. Utilizando este conjunto, os sinais foram filtrados com os algoritmos de Wiener convencional (AWC) e algoritmo de Wiener proposto (AWP). Para cada sinal de voz, variamos o parâmetro $A_{\max}$ no intervalo $0,5 < A_{\max} < 2,5$ e aplicamos o filtro com o respectivo parâmetro. Com esses resultados, aplicamos a cada sinal os algoritmos CSII e PESQ para obtermos os respectivos índices. Finalmente, foi feita a média dos índices para cada caso.

Apesar do procedimento ser simples, o esforço computacional é bastante elevado. Elevado não somente pelo cálculos efetuados pelos algoritmos em si, mas principalmente pelos cálculos executados pelas rotinas matemáticas que calculam os índices CSII e PESQ. Somando a isso, ainda foram feitas as médias e desvios para cada sinal de voz utilizando todas as combinações dos parâmetros variáveis. Os programas em Matlab encontram-se no Anexo

A e os detalhes de implementação no Anexo B.

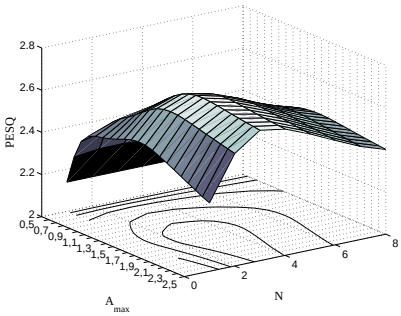
3.6.1 Ruído branco

Os resultados reportados na Seção 3.5.1 mostram uma baixa sensibilidade do PESQ em relação ao valor de σ_g. O mesmo, porém, não pode ser dito sobre a sensibilidade de $A_{\max}$ em relação ao valor de N. Para melhor estudarmos a influência de N sobre $A_{\max}$ e, consequentemente, sobre a qualidade de voz e a inteligibilidade, traçamos as superfícies de desempenho utilizando os indicadores PESQ e CSII como funções de N e $A_{\max}$. Essas superfícies são mostradas, respectivamente, nas Figuras 8 e 9.

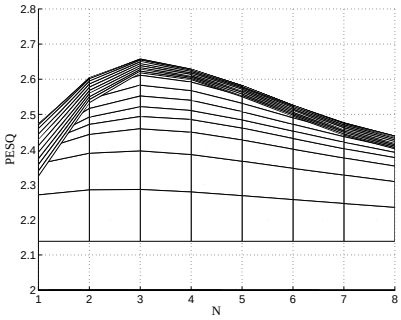
A Figura 8(a) mostra a vista tri-dimensional da superfície de desempenho gerada com o indicador PESQ para SNR=2 dB. A partir da vista lateral apresentada na Figura 8(b), podemos facilmente determinar o valor de N que maximiza o PESQ nesse caso, que é $N = 3$. Observando a Figura 8(c), concluímos que para este valor N o valor correspondente para A_{max_o} é de $A_{\max} \approx 1,5$. Verificamos experimentalmente que esses valores ótimos não mudam significativamente para SNRs variando no intervalo de -10 dB a 20 dB.

A Figura 9(a) mostra a vista tri-dimensional da superfície gerada com o indicador PESQ para SNR=2 dB. A Figura 9(b) mostra que a inteligibilidade é maximizada com $A_{\max} = 0,8$. Procurando por esse valor na Figura 9(c), vemos que ao longo desta linha a máxima inteligibilidade é dada por $N = 3$. É importante salientar que $A_{\max} = 0,5$ representa tanto o nível de inteligibilidade (CSII) quanto a qualidade (PESQ) atingida pelo AWC, o qual utiliza um valor constante $\beta = 0,5$.

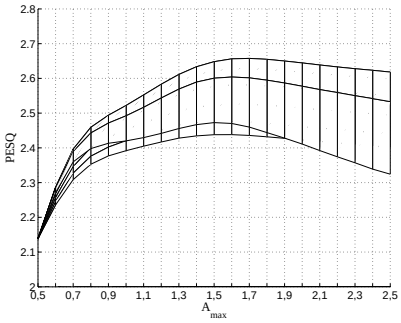
Combinando os resultados das Figuras 8 e 9, verificamos que há uma relação de compromisso no projeto destes parâmetros. Entretanto, mesmo com o valor de $A_{\max} = 1,5$ sugerido pela maximização do PESQ, juntamente com $N = 3$, o valor correspondente de CSII será equivalente àquele prove-



(a) Vista 3D



(b) Vista lateral direita



(c) Vista lateral esquerda

Figura 8: Superfície de desempenho para ruído branco utilizando o PESQ (SNR = 2 dB)

niente do algoritmo convencional (ver Figura 9(b)). Portanto, qualquer $A_{\max}$ no intervalo $(0,5;1,5)$ resultará em uma inteligibilidade tão boa quanto, ou melhor do que aquela obtida utilizando o algoritmo Wiener convencional. Embora as superfícies mostradas sejam para SNR = 2 dB, verificamos que esses valores dos parâmetros de projeto levam a resultados semelhantes [4] para SNRs no intervalo descrito neste estudo.

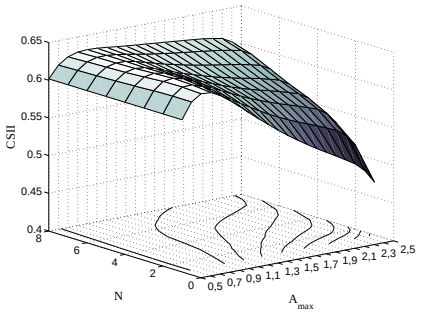
3.6.2 Ruídos reais

Até agora consideramos somente sinais de voz corrompidos por ruído branco. Para esse caso, o algoritmo proposto provou ter um bom desempenho. Uma avaliação mais realista requer a aplicação do algoritmo proposto a sinais de voz corrompidos por ruídos reais. Para isto, usamos vários tipos de ruído, como por exemplo ruído de restaurante, ruído de ventilador, ruído de uma estação de trem, ruído de rua movimentada e também ruído de aspirador de pó. Como as superfícies de desempenho dos diversos casos têm comportamentos semelhantes, como exemplo a superfície para ruído de ventilador. As demais superfícies são apresentadas no Anexo D.

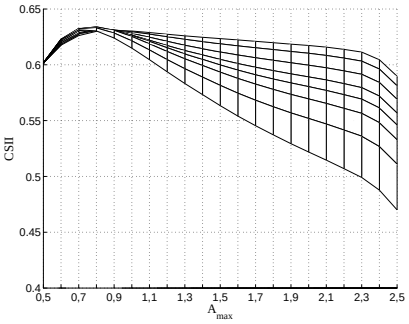
3.6.2.1 Ruído de ventilador

Analisando os sinais corrompidos com ruído de ventilador, podemos ver que as superfícies de desempenho apresentam comportamento semelhante ao do caso do ruído branco. Considerando a superfície de desempenho utilizando PESQ, podemos concluir que o valor ótimo para o número de harmônicas é $N = 3$ como pode ser observado na Figura 10(b). A Figura 10(c) mostra a vista lateral da superfície que evidencia a maximização de $A_{\max}$. Conforme mostrado nesta vista, podemos concluir que o valor ótimo é $A_{\max_o} \approx 1,5$. O comportamento mais geral do algoritmo proposto, considerando o PESQ, é apresentado através da vista tri-dimensional na Figura 10(a).

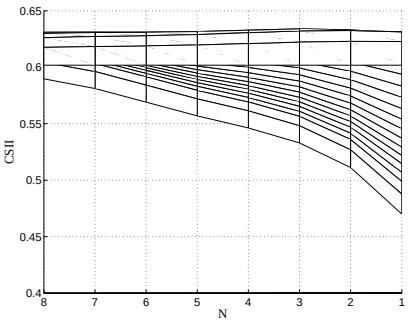
[4] Uma comparação para outras SNRs está presente na Seção 3.7



(a) Vista 3D



(b) Vista lateral direita



(c) Vista lateral esquerda

Figura 9: Superfície de desempenho para ruído branco utilizando o CSII (SNR = 2 dB)

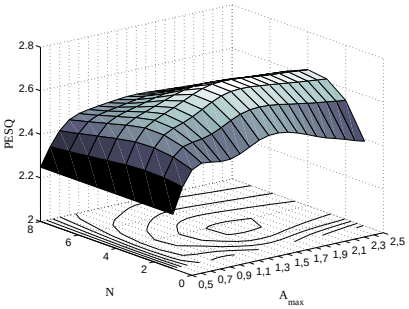
Agora, faremos uma análise da inteligibilidade através da superfície de desempenho utilizando o CSII. Esta análise será feita utilizando a Figura 11. A Figura 11(a) apresenta o comportamento geral do algoritmo proposto através da vista tri-dimensional. Podemos concluir através da Figura 11(c) que CSII apresenta valor máximo em $A_{\max_o} = 0,8$. Na Figura 11(c) vemos que o número de harmônicas ótimo é $N = 3$. Com esses resultados, vemos que os valores de $A_{\max_o}$ é diferente para CSII e PESQ. Então, devemos escolher um valor intermediário para obtermos o melhor resultado de inteligibilidade e qualidade de voz.

3.7 Comparação dos melhores casos

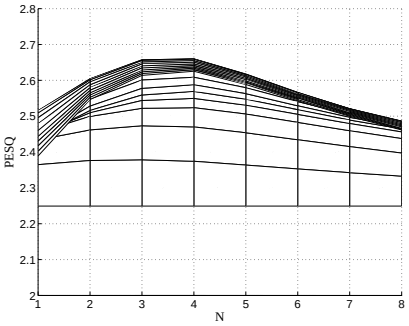
As superfícies de desempenho nos deram um boa noção do comportamento do algoritmo ao alterarmos os parâmetros de controle. Através delas pudemos concluir que na média $N = 3$ é o melhor parâmetro para o número de harmônicas.

Para fazermos uma análise mais direcionada, agrupamos todas as curvas com $N = 3$ com diferentes SNRs para cada tipo de ruído abordado no estudo. Para deixar mais claro, vamos pegar como exemplo a Figura 12(a) referente ao ruído de ventilador. Para cada SNR estudada, selecionamos as curvas com $N = 3$ e agrupamos em um único gráfico que mostra como $A_{\max}$ varia de acordo a SNR. O mesmo se aplica aos demais ruídos e estende-se aos indicadores PESQ e CSII. Essas curvas são mostradas nas Figuras 12 e 13.

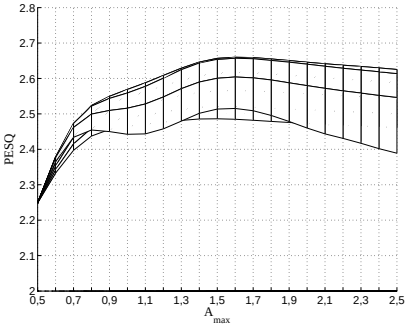
Podemos ver na Figura 12 que as curvas atingem seus máximos em torno de $A_{\max} = 0,8$. O mesmo vale para o indicador PESQ (Figura 13) no intervalo de 20 dB a 10 dB. No entanto, abaixo de 2 dB, as curvas do indicador PESQ atingem $A_{\max_o}$ em valores bem maiores. Além disso, o comportamento destas curvas varia dependendo da SNR. Em SNRs altas o intervalo onde há de ganho em inteligibilidade é mais estreito. Por exemplo, $A_{\max} \approx 0,8$, mas a medida que a SNR tende a valores menores, o intervalo de ganho em inte-



(a) Vista 3D

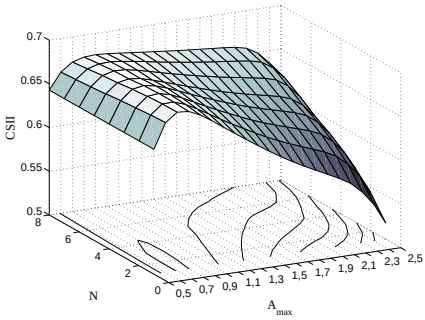


(b) Vista lateral direita

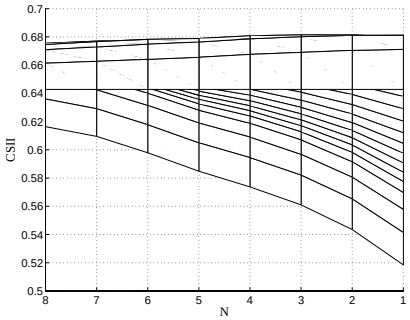


(c) Vista lateral esquerda

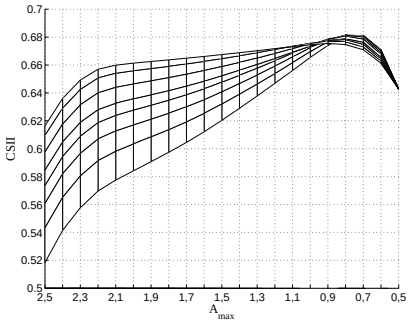
Figura 10: Superfície de desempenho para ruído de ventilador utilizando o PESQ (SNR = 2 dB)



(a) Vista 3D



(b) Vista lateral direita



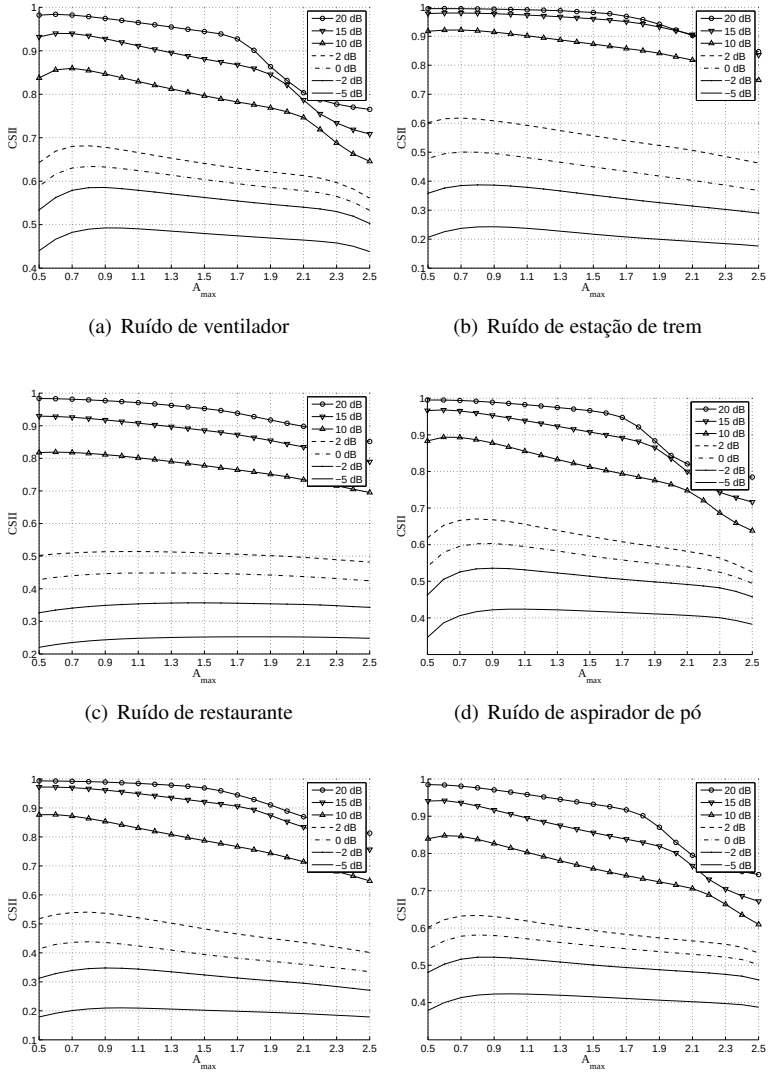
(c) Vista lateral esquerda

Figura 11: Superfície de desempenho para ruído de ventilador utilizando o CSII (SNR = 2 dB)

ligibilidade é mais amplo e inicia em um valor de $A_{\max}$ um pouco maior. Em SNRs baixas, quanto maior o valor de $A_{\max}$ maior será a atenuação do filtro, levando a uma boa diminuição do ruído residual no sinal filtrado. Isto se deve ao fato do filtro de Wiener paramétrico elevar a atenuação quando a SNR é baixa e reduzir quando a SNR é alta. Embora para SNRs acima de 10 dB o ganho do filtro de Wiener tenda para 0 dB, ainda há uma pequena variação de ganho. Somado a isso, cada quadro possui em média um valor fixo, porém, para cada bin de frequência há uma SNR maior ou menor, levando a atenuações diferentes, ou seja, diferentes pontos em cima da curva de atenuação (ver Figura 2(b)). Essa soma de fatores se traduz em uma variação de qualidade bem visível no PESQ. Subjetivamente, percebemos uma boa melhora até $A_{\max} = 1$ e a partir daí o sinal começa a sofrer distorções. O que podemos concluir é que embora o gráfico do PESQ, para SNR elevadas, indique $A_{\max_o} > 1$, testes subjetivos provam que acima deste valor a inteligibilidade é prejudicada. Como estamos focando na inteligibilidade, $A_{\max}$ deve ser menor que 1, como podemos verificar pelo indicador CSII (ver Figura 12). Os casos apresentados são uma média entre os sinais de voz masculina e feminina, no entanto, podemos ver separadamente cada um dos casos no Anexo C.

O valor de $A_{\max_o}$ foi determinado após diversos experimentos subjetivos. Notamos em testes subjetivos que ao exceder o valor de $A_{\max_o}$ com o índice CSII, há considerável perda de inteligibilidade. Como nosso foco é maximizar a inteligibilidade, usaremos $A_{\max} = 0,8$. Apesar desse valor não ser o $A_{\max_o}$ pelo índice PESQ, utilizando $A_{\max} = 0,8$ obtemos um grande ganho em qualidade subjetiva em relação ao AWC. Baseado nos testes, concluímos que um bom compromisso entre qualidade e inteligibilidade é obtido com $A_{\max} = 0,8$, $A_{\min} = 0,5$ e $N = 3$. Esses valores serão usados nos testes estatísticos da Seção 4.1, exceto quando mencionado o contrário.

Neste capítulo apresentamos uma modificação do parâmetro β para o filtro de Wiener paramétrico. Normalmente β é um escalar e é empregado



(a) Ruído de ventilador

(b) Ruído de estação de trem

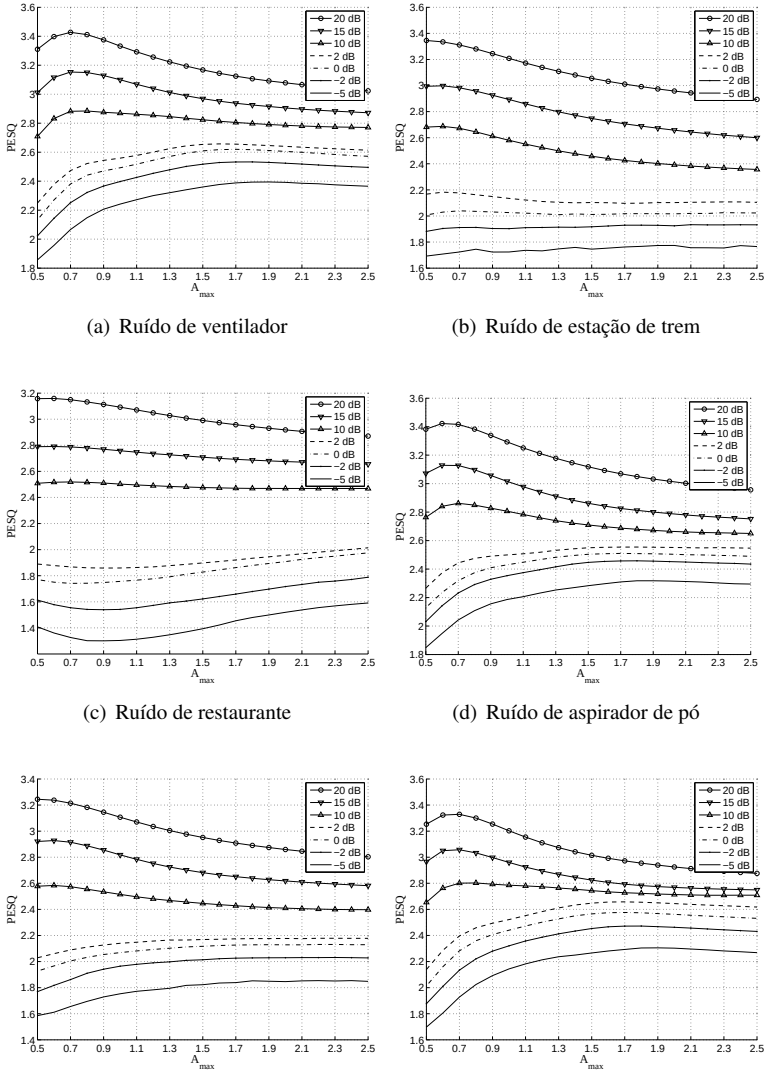
(c) Ruído de restaurante

(d) Ruído de aspirador de pó

(e) Ruído de rua movimentada

(f) Ruído branco

Figura 12: Comparação do efeito de $A_{\max}$ para diferentes SNRs ($N = 3$), utilizando CSII.



(a) Ruído de ventilador

(b) Ruído de estação de trem

(c) Ruído de restaurante

(d) Ruído de aspirador de pó

(e) Ruído de rua movimentada

(f) Ruído branco

Figura 13: Comparação do efeito de $A_{\max}$ para diferentes SNRs ($N = 3$), utilizando PESQ.

para toda a banda de frequência e constante para todos os quadros de voz processados. Como a modificação proposta, β é agora uma função *pitch* e com isso é possível empregar uma atenuação diferente para cada bin de frequência que varia de quadro para quadro. Resultados preliminares mostraram a potencialidade na nova técnica para diferentes tipos de ruído e SNRs.

No próximo capítulo, faremos uma análise mais detalhada de desempenho do algoritmo proposto utilizando testes estatísticos. A avaliação estatística é primordial para garantir que, na média, teremos resultados superiores ao algoritmo convencional. Esta avaliação será baseada nos indicadores PESQ e CSII através do teste-t.

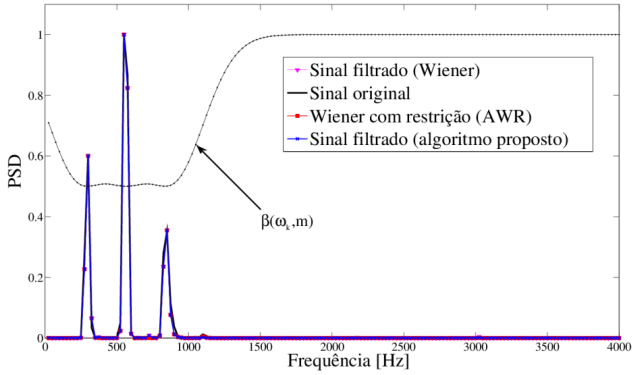
4 RESULTADOS

Como referência para máxima inteligibilidade, nós aplicamos a restrição (2.54) ao algoritmo de Wiener usando (3.1). Este algoritmo será referido como Algoritmo de Wiener com Restrição (AWR).

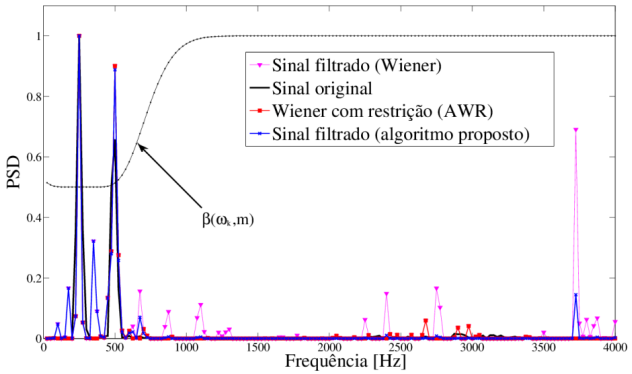
A Figura 14(a) representa o espectro de potência de um quadro "ideal", com três picos grandes nas três primeiras harmônicas e baixa energia no restante do espectro. A função de atenuação $\beta(\omega_k, m)$ com $A_{\min} = 0,5$ e $A_{\max} = 1,0$ também é mostrada. Nesse caso, tanto o algoritmo tradicional quanto o proposto apresentam resultados semelhantes como era de se esperar, já que não há componentes indesejadas em altas frequências. A Figura 14(b) mostra um outro quadro mais comum no qual o algoritmo de Wiener convencional preserva componentes errôneos de alta frequência devido a sua baixa atenuação nessas frequências. O algoritmo proposto, por outro lado, proporciona uma atenuação mais elevada para as componentes indesejáveis em altas frequências. Idealmente, esses picos indesejáveis seriam totalmente removidos se (2.54) pudesse ser usado. O método proposto remove ou atenua drasticamente esses picos, levando a resultados práticos muito próximos àqueles obtidos utilizando (2.54). Como o algoritmo proposto emprega uma atenuação mais elevada nas frequências altas, é possível que alguma informação importante seja perdida. Por outro lado, a remoção deste picos aleatoriamente espaçados que mudam de lugar de quadro para quadro proporciona uma redução do ruído musical e também um aumento da inteligibilidade.

4.1 Avaliação do desempenho estatístico

A avaliação estatística do desempenho foi conduzida aplicando as medidas objetivas PESQ e CSII nos 48 sinais de voz pertencentes ao CV. Essas medidas foram aplicadas com o objetivo de comparar os resultados do pro-



(a) Quadro m_1.



(b) Quadro m_2.

Figura 14: Espectro de potência de dois quadros de voz m_1 and m_2 aleatoriamente selecionados.

cessamento dos com o algoritmo convencional e com o algoritmo proposto. Os parâmetros utilizados no algoritmo proposto foram $N = 3$ e $A_{\max} = 0.8$, com base nas conclusões obtidas no Capítulo 3. Dentre os testes estatísticos disponíveis para comparar conjuntos de dados, o teste-t foi utilizado por sua simplicidade, confiabilidade e também por ser recomendado pela ITU-T P.835 (ITU-T, 2003). Os resultados são baseados em um teste-t unilateral pareado a 5 % de significância. Foram testados os dois casos, à direita e à esquerda para verificar tanto um ganho significativo (indicado pelo símbolo "+"), quanto uma perda significativa (indicado pelo símbolo "-") de desempenho. Os casos em que não há ganho nem perda significativa, são usado o indicado pelo símbolo "=". Os resultados foram obtidos utilizando todos os sinais de voz. Portando, obtivemos valores de PESQ e CSII médios considerando sinais de voz masculina e feminina.

A avaliação foi dividida em duas partes, sendo elas separadas pelo tipo de VAD utilizado. Na primeira parte, faremos uma avaliação estatística com o mesmo VAD ideal utilizado nas superfícies de desempenho para mostrar um resultado de referência, ou seja, quão bem o algoritmo proposto irá desempenhar se tivermos um VAD que funcione sem erros. Na segunda parte, utilizaremos um VAD real, o bem conhecido VAD baseado em modelo estatístico (SOHN; KIM; SUNG, 1999), para detectar a presença de voz tanto no algoritmo convencional, como no algoritmo proposto. Adotaremos como valores padrão para os testes estatísticos realizados os parâmetros $N = 3$ e $A_{\max} = 0,8$, exceto quando valores diferentes forem explicitamente especificados.

4.1.1 Escolha do teste estatístico

Para avaliar o desempenho do algoritmo proposto, foi feita uma análise estatística com as pontuações obtidas pelo PESQ, tanto para o algoritmo Wiener tradicional quanto para o proposto. O teste estatístico é necessário para avaliar se houve um ganho significativo em desempenho quando compa-

rado ao algoritmo convencional.

Há uma grande quantidade de testes estatísticos para esse fim. Um teste frequentemente empregado, e que é sugerido na recomendação ITU-T P.835 (ITU-T, 2003) é o teste-t. Um caso especial do teste-t para duas amostras é o teste-t pareado. Isto ocorre quando as duas populações de interesse são coletadas em pares. Ou seja, cada par de observações é gerado sob condições homogêneas, mas essas condições podem variar de um par para o outro (MONTGOMERY; RUNGER, 2003). Isto ocorre, por exemplo, quando as diferentes populações resultam da aplicação de algoritmos diferentes a um mesmo sinal ruidoso. Portanto, o procedimento do teste consiste em analisar as diferenças das médias das pontuações PESQ/CSII.

Dentre os cenários possíveis temos: Algoritmos proposto e convencional não apresentam diferença significativa, Algoritmo proposto significativamente melhor do que o algoritmo tradicional e vice-versa. Para avaliar esses cenários vamos primeiramente avaliar se há diferença estatística significativa através do teste-t bilateral, onde μ_1 representa a média da pontuação PESQ/CSII para o algoritmo convencional e μ_2 representa a média de pontuação PESQ/CSII para o algoritmo proposto:

$$H_1 : \mu_1 = \mu_2 \tag{4.1}$$

$$H_0 : \mu_1 \neq \mu_2, \tag{4.2}$$

no entanto, esse teste somente nos dá a informação se os algoritmos são significativamente diferentes ou não.

Como queremos determinar se o algoritmo proposto é significativamente melhor do que o convencional, devemos utilizar o teste unilateral e utilizar as hipóteses

$$H_1 : \mu_1 - \mu_2 < 0 \tag{4.3}$$

$$H_0 : \mu_1 - \mu_2 = 0 \tag{4.4}$$

Assim utilizaremos a parte esquerda do teste unilateral para verificar se a média de pontuação PESQ/CSII para o algoritmo proposto é significativamente maior que o algoritmo convencional.

Para verificar se a média de pontuação PESQ/CSII para o algoritmo proposto é significativamente menor que o algoritmo convencional utilizamos as hipóteses:

$$H_1 : \mu_1 - \mu_2 > 0 \tag{4.5}$$

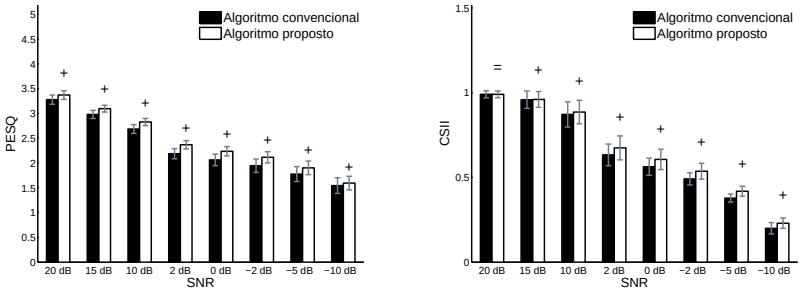
$$H_0 : \mu_1 - \mu_2 = 0. \tag{4.6}$$

4.2 VAD ideal

Os resultados apresentados a seguir foram gerados com os sinais processados pelos algoritmos proposto e convencional utilizando o VAD ideal a fim de estabelecermos uma referência de desempenho quando o VAD real for aplicado na seção posterior. Os teste foram realizados com diferentes tipos de ruídos para uma melhor caracterização do desempenho dos algoritmos.

4.2.1 Ruído branco

Os resultados do teste-t para ruído branco são mostrados na Figura 15 para os indicadores PESQ e CSII. Conforme podemos ver, os resultados indicam ganho tanto em qualidade do sinal quanto em inteligibilidade para praticamente todas as SNRs testadas. A única exceção ocorre para o valor do CSII em 20 dB. Nesse caso, a diferença de desempenho para ambos os algoritmos é estatisticamente insignificante.



(a) PESQ (b) CSII

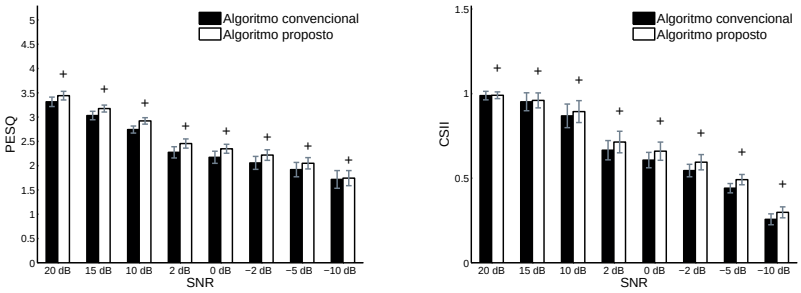
Figura 15: Teste-t para sinais de voz masculina e feminina corrompidos por ruído branco gaussiano filtrados pelos algoritmos AWC e AWP, utilizando os indicadores PESQ (a) e CSII (b).

4.2.2 Ruído de ventilador

A Figura 16 mostra os resultados do teste-t para ruído de ventilador. Como a superfície de desempenho nesse caso é muito parecida com a superfície do caso de ruído branco, espera-se que os testes estatísticos também indiquem melhorias significativas. É fácil de ver que obtivemos melhorias tanto em inteligibilidade quanto em qualidade do sinal, como esperado. Podemos ver que os valores do CSII em 20 dB apresentam o mesmo valor médio. No entanto, como o desvio é menor conseguimos um resultado melhor.

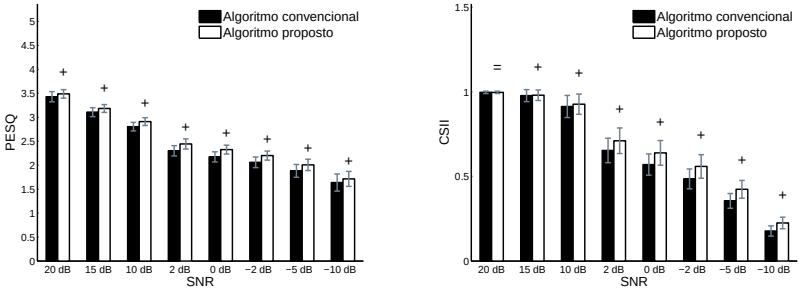
4.2.3 Ruído de aspirador de pó

A Figura 17 mostra os resultados para para ruído de aspirador de pó. Através dela, podemos verificar uma melhora significativa em todos os casos, exceto para o CSII em 20 dB. Para essa SNR, os dois algoritmos considerados apresentam desempenhos estatisticamente semelhantes.



(a) PESQ (b) CSII

Figura 16: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de ventilador filtrados pelos algoritmos AWC e AWP, utilizando os indicadores PESQ (a) e CSII (b).

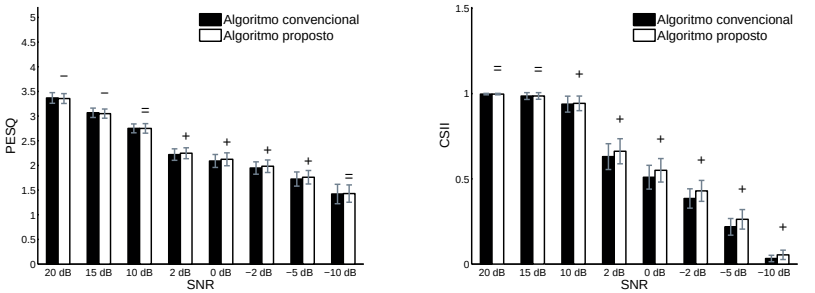


(a) PESQ (b) CSII

Figura 17: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de aspirador de pó filtrados pelos algoritmos AWC e AWP, utilizando os indicadores PESQ (a) e CSII (b).

4.2.4 Ruído de estação de trem

Analisando a Figura 18(a), podemos ver que na faixa de 2 dB a -5 dB o algoritmo proposto levou a um melhor desempenho pelo índice PESQ. Os resultados foram ligeiramente piores para 20 dB e 15 dB, e equivalentes para as demais SNRs. Já a Figura 18(b) mostra que o algoritmo proposto levou a uma grande melhora na inteligibilidade, exceto para para 20 dB e 15 dB, casos em que o desempenho foi equivalente. Através de testes subjetivos concluímos que mesmo nos casos em que a qualidade (medida pelo PESQ) é inferior, os sinais são mais agradáveis de escutar. Acreditamos que isso seja devido à menor quantidade de ruído musical.



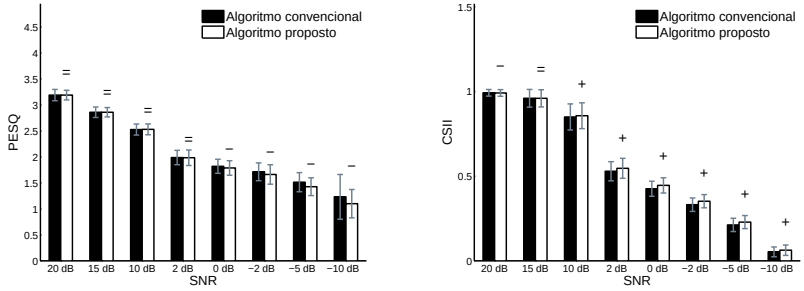
(a) PESQ (b) CSII

Figura 18: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de estação de trem filtrados pelos algoritmos AWC e AWP, utilizando os indicadores PESQ (a) e CSII (b).

4.2.5 Ruído de restaurante

Analisando a Figura 19(a) notamos desempenho equivalente no intervalo de $[20, 2]$ dB. Porém, fora deste intervalo o desempenho do algoritmo proposto foi ligeiramente inferior. Apesar desses resultados negativos, a inteligibilidade é aumentada significativamente entre $[10, -10]$ dB e é equivalente

em 20 dB e 10 dB, conforme podemos observar na Figura 19(b). Subjetivamente, podemos notar nitidamente a melhora na inteligibilidade, bem como a redução do ruído musical.



(a) PESQ (b) CSII

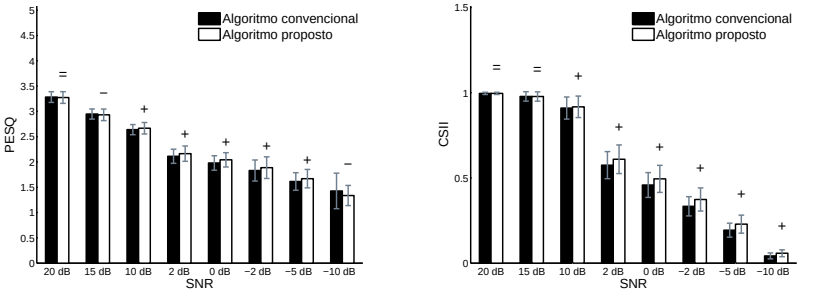
Figura 19: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de restaurante filtrados pelos algoritmos AWC e AWP

4.2.6 Ruído de rua movimentada

Analisando a Figura 20(a), podemos ver que a qualidade do sinal processado com o algoritmo proposto ficou significativamente melhor ou igual ao algoritmo convencional exceto para 15 dB e -10 dB. Os algoritmos tiverem desempenhos equivalentes para 20 dB e 15 dB, e para as demais SNRs a inteligibilidade proporcionada pelo novo algoritmo foi bem melhor. Esses resultados foram comprovados subjetivamente.

4.3 VAD real

Os resultados apresentados na seção anterior indicam o potencial do algoritmo proposto, uma vez que foram obtidos usando um VAD ideal. Nesta seção apresentaremos os resultados na situação mais prática, ou seja, com um VAD real. Os resultados são apresentados da Figura 21 até a Figura 26 para



(a) PESQ (b) CSII

Figura 20: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de rua filtrados pelos algoritmos AWC e AWP

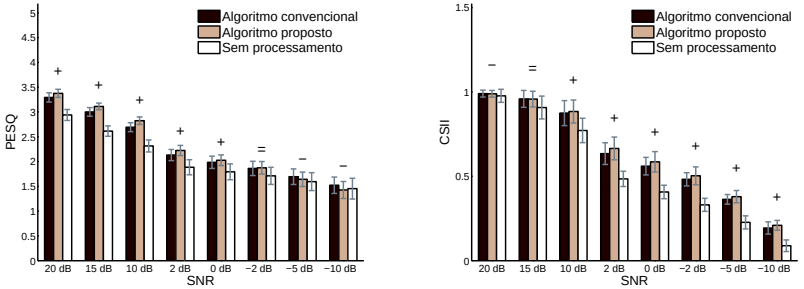
os mesmos ruídos tratados anteriormente. Comparando cada caso desta seção com os respectivos casos da Seção 4.2, notamos que o bom funcionamento do VAD é um fator importante no desempenho do algoritmo proposto, principalmente em baixa SNR. Apesar de o algoritmo proposto geralmente apresentar um desempenho inferior (quando comparado com o caso do VAD ideal) na qualidade do sinal, a inteligibilidade quase não é afetada, levando a resultados tão bons quanto no caso ideal. Obviamente, a qualidade do sinal é importante. Porém, mesmo com a qualidade afetada negativamente pelo desempenho do VAD, notamos uma maior facilidade de compreender as sentenças pronunciadas nos testes subjetivos. De qualquer modo, a maior parte dos algoritmos de redução de ruído sofre com o baixo desempenho dos VADs em baixa SNR e com a presença de ruídos não estacionários. Com a evolução dos algoritmos de detecção de voz, a tendência será a de melhoria de desempenho dos algoritmos de redução de ruído, incluindo o proposto neste trabalho.

Nesta seção, fizemos a análise estatística incluindo também os sinais de voz ruidosos sem qualquer processamento. Isso foi feito para termos os escores de PESQ e CSII também para os sinais sem processamento para servir

de referência. Esses dados adicionais são importantes, pois com eles é possível determinar se realmente vale a pena aplicar os algoritmos de redução de ruído nos sinais ruidosos (para uma determinada SNR) ou deixá-los sem qualquer processamento. Esta análise é feita para o indicador PESQ como também para o indicador CSII.

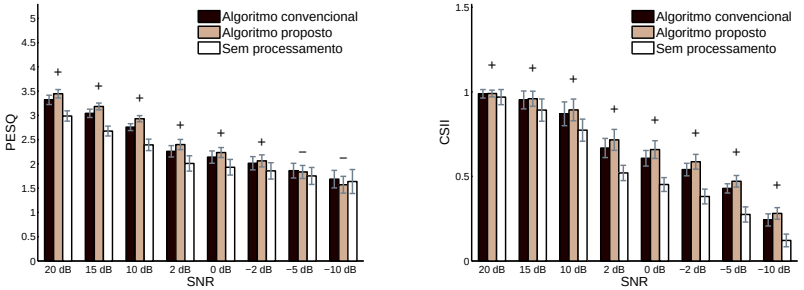
Esta análise foi feita em dois pares: Sinais sem processamento juntamente com AWC e sinais sem processamento com AWP. Ou seja, primeiramente aplicamos o teste-t aos sinais sem processamento juntamente com os sinais aplicando o AWC. Posteriormente, aplicamos o teste-t aos sinais sem processamento juntamente com os sinais aplicando o AWP. Desta forma, quando for estatisticamente melhor (5 % de significância) deixar os sinais sem qualquer processamento, ou seja, sem aplicar qualquer técnica de redução de ruído (AWC ou AWP), este caso será identificado com uma flecha em cima de cada barra, para a SNR e tipo de ruído aditivo considerado.

Conforme podemos observar, há somente um caso onde é estatisticamente melhor deixar os sinal sem processamento: ruído de restaurante. Mesmo assim, isso só verdade considerando o indicador PESQ e SNR abaixo de 2 dB. Entretanto, considerando a inteligibilidade, é sempre melhor aplicar o AWC ou AWP aos sinal ruidosos. Isso é coerente com a análise subjetiva feita em laboratório, onde foi possível perceber um ganho de inteligibilidade para todas as SNRs consideradas nesse trabalho.



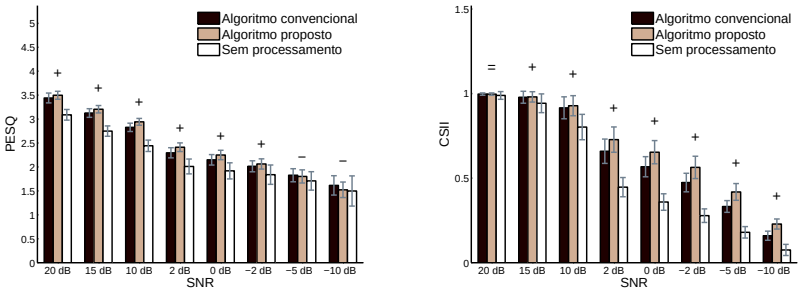
(a) PESQ (b) CSII

Figura 21: Teste-t para sinais de voz masculina e feminina corrompidos por ruído branco gaussiano filtrados pelos algoritmos AWC e AWP



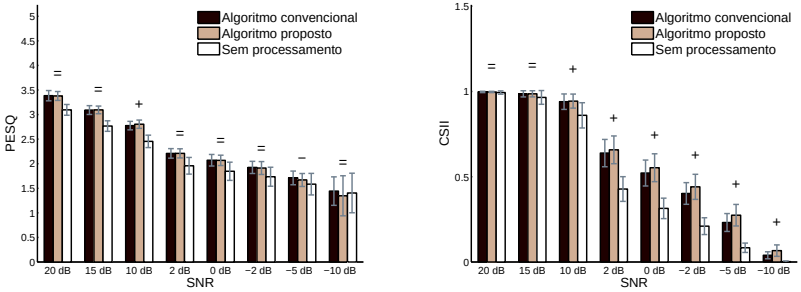
(a) PESQ (b) CSII

Figura 22: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de ventilador filtrados pelos algoritmos AWC e AWP



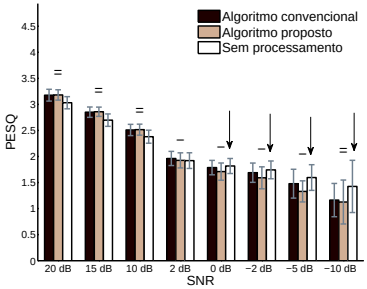
(a) PESQ (b) CSII

Figura 23: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de aspirador de pó filtrados pelos algoritmos AWC e AWP

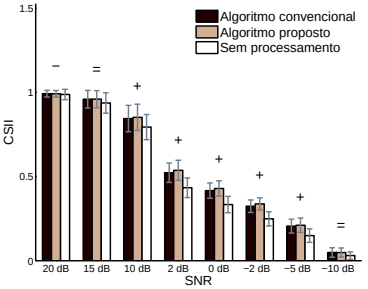


(a) PESQ (b) CSII

Figura 24: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de estação de trem filtrados pelos algoritmos AWC e AWP

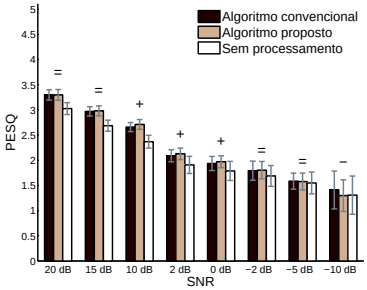


(a) PESQ

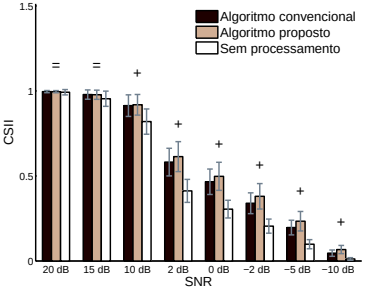


(b) CSII

Figura 25: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de restaurante filtrados pelos algoritmos AWC e AWP



(a) PESQ



(b) CSII

Figura 26: Teste-t para sinais de voz masculina e feminina corrompidos por ruído de rua filtrados pelos algoritmos AWC e AWP

5 CONCLUSÃO

Este trabalho propôs uma modificação na função de ganho do filtro de Wiener para redução de ruído em sinais de voz. Um novo parâmetro adaptativo dependente da frequência (função do *pitch*) foi proposto para substituir o parâmetro β constante empregado no algoritmo convencional. O algoritmo proposto enfatiza as componentes espectrais em torno do *pitch* e de suas primeiras $N-1$ harmônicas. As componentes de frequência distantes do *pitch* e de seus múltiplos recebem maior atenuação. Em comparação com o algoritmo convencional utilizando testes estatísticos, o algoritmo proposto apresentou uma melhora significativa de 5% nos indicadores PESQ e também CSII para a maioria dos casos testados.

Testes feitos utilizando um VAD ideal servem de referência teórica e de indicativo da potencialidade do novo algoritmo. Nestes testes, a inteligibilidade teve desempenho significativamente superior para a maioria das SNRs consideradas. Obviamente, dependendo do ruído aditivo presente nos sinais de voz tivemos desempenhos um pouco diferenciados. Quando utilizamos um VAD real, houve uma perda de desempenho esperada em praticamente todos os casos. Porém, na maioria dos casos ainda conseguimos resultados significativamente superiores aos obtidos usando o algoritmo de Wiener convencional.

Notamos ainda que encontrar um meio termo entre qualidade e inteligibilidade é uma tarefa difícil. No entanto, em muitas aplicações a inteligibilidade é de suma importância. Nessas aplicações, geralmente é aceitável sacrificar um pouco a qualidade se a inteligibilidade puder ser melhorada. Em nosso trabalho, buscamos fazer exatamente isso, ou seja, encontrar o melhor conjunto de parâmetros para o algoritmo proposto de tal forma que a inteligibilidade seja maximizada.

A partir das superfícies de desempenho, foi possível selecionar um nú-

mero médio de harmônicas N tal que o desempenho médio seja maximizado no sentido da inteligibilidade. Uma melhoria de desempenho pode ainda ser obtida com uma melhor estimação do número ideal de harmônicas a serem mantidas para cada quadro.

5.1 Sugestões para continuação do trabalho

Como vimos, o algoritmo proposto no presente trabalho provou ter um bom ganho tanto em inteligibilidade quanto em qualidade de voz. Conseguimos esse objetivo aplicando um ganho diferente na parte do espectro do sinal onde se encontra a maior parte da energia relevante do sinal de voz. No entanto, fizemos isso selecionando um número médio das harmônicas de cada quadro de voz processado.

Esse resultado pode ser melhorado ainda mais selecionando um número diferente de harmônicas para cada quadro. Esse número pode variar de duas a três harmônicas ao longo de cada quadro do sinal. Se essa distinção for feita para cada quadro, por exemplo através de um limiar ou de um estimador, podemos reduzir ainda mais as distorções inseridas no sinal voz e filtrar ainda mais as harmônicas indesejáveis que contém ruído.

Além disso, como mencionado ao longo do trabalho, para quadros não-vozeados o algoritmo que estima o valor do *pitch* retorna uma valor bem pequeno, fazendo com que $\beta \rightarrow 1$ para a maior parte do espectro no quadro processado. Se um discriminador de quadros vozeados/não vozeados for utilizado, podemos empregar um valor de β menor, como por exemplo 0,5. Esse seria o valor ideal, já que esse é o caso especial do filtro de Wiener paramétrico e possui boas propriedades espectrais para o sinal.

REFERÊNCIAS

AJIBOYE, B. *Matlab Central: Plots a fully customizable grouped bar graph with error bars @ONLINE*. 2006. Disponível em: <http://www-.mathworks.com/matlabcentral/fileexchange/10803>.

ALAM, M. J.; O'SHAUGHNESSY, D. Perceptual improvement of wiener filtering employing a post-filter. *Digital Signal Processing: A Review Journal*, v. 21, n. 1, p. 54 – 65, 2010. ISSN 10512004.

AMEHRAYE, A.; PASTOR, D.; TAMTAOUI, A. Perceptual improvement of wiener filtering. In: . Las Vegas, NV, United states: [s.n.], 2008. p. 2081 – 2084. ISSN 15206149.

AREHART, K. H. et al. Effects of noise and distortion on speech quality judgments in normal-hearing and hearing-impaired listeners. *The Journal of the Acoustical Society of America*, ASA, v. 122, n. 2, p. 1150–1164, 2007.

BEERENDS, J. G. et al. Measurement of speech intelligibility based on the pesq approach. In: *Measurement of Speech, Audio and Video Quality in Networks (ME-SAQIN)*. Prague, Czech Republic: [s.n.], 2004.

BEERENDS, J. G.; WIJNGAARDEN, S. van; BUUREN, R. van. Extension of itu-t recommendation p.862 pesq towards measuring speech intelligibility with vocoders. In: NATO RESEARCH AND TECHNOLOGY ORGANISATION. *RTO-MP-HFM-123 - New Directions for Improving Audio Effectiveness*. Prague, Czech Republic, 2005. p. 10–1 – 10–6.

BERNSTEIN, D. S. *Matrix mathematics: Theory, facts, and formulas with application to linear systems theory*. Princeton, NJ, USA: Princeton University Press, 2005.

BEROUTI, M.; SCHWARTZ, R.; MAKHOUL, J. Enhancement of speech corrupted by acoustic noise. Washington, DC, USA, p. 208 – 211, 1979.

BOLL, S. Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 27, n. 2, p. 113 – 120, apr. 1979. ISSN 0096-3518.

CAMACHO, A. Detection of pitched/unpitched sound using pitch strength clustering. In: *ISMIR*. [S.l.: s.n.], 2008. p. 533–537.

CAPPE, O. Elimination of the musical noise phenomenon with the ephraim and malah noise suppressor. *IEEE Transactions on Speech and Audio Processing*, New York, NY, United States, v. 2, n. 2, p. 345 – 349, 1994. ISSN 10636676.

CHEHRESA, S.; SAVOJI, M. Codebook constrained iterative and parametric wiener filter speech enhancement. In: *Signal and Image Processing Applications (ICSIPA), 2009 IEEE International Conference on*. [S.l.: s.n.], 2009. p. 548 –553.

CHEN, F.; LOIZOU, P. C. Speech enhancement using a frequency-specific composite wiener function. In: . [S.l.: s.n.], 2010. p. 4726 –4729. ISSN 1520-6149.

DING, H. et al. A post-processing technique for regeneration of over-attenuated speech components. In: . Taipei, Taiwan: [s.n.], 2009. p. 3889 – 3892. ISSN 15206149.

EPHRAIM, Y.; MALAH, D. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32, n. 6, p. 1109 – 1121, 1984. ISSN 00963518.

EPHRAIM, Y.; TREES, H. V. A signal subspace approach for speech enhancement. *Speech and Audio Processing, IEEE Transactions on*, v. 3, n. 4, p. 251 –266, jul. 1995. ISSN 1063-6676.

ESCH, T.; VARY, P. Efficient musical noise suppression for speech enhancement systems. In: . Taipei, Taiwan: [s.n.], 2009. p. 4409 – 4412. ISSN 15206149.

HANSEN, J. H. L. *Analysis and compensation of stressed and noisy speech with application to robust automatic recognition*. Tese (Doutorado) — Georgia Institute of Technology, Atlanta, GA, USA, 1988.

HANSEN, J. H. L.; CLEMENTS, M. A. Constrained iterative speech enhancement with application to speech recognition. *IEEE Transactions on Signal Processing*, 1991.

HAYKIN, S. *Adaptive Filter Theory (4th Edition)*. [S.l.]: Prentice Hall, 2002. ISBN 0130901261.

HIMMELBLAU, D. M. Book. *Applied nonlinear programming [by] David M. Himmelblau*. [S.l.]: McGraw-Hill New York,, 1972. xi, 498 p. p. ISBN 0070289212 0070289212.

HU, Y.; LOIZOU, P. Evaluation of objective quality measures for speech enhancement. *Audio, Speech, and Language Processing, IEEE Transactions on*, v. 16, n. 1, p. 229 –238, jan. 2008. ISSN 1558-7916.

HU, Y.; LOIZOU, P. C. Incorporating a psychoacoustical model in frequency domain speech enhancement. *IEEE Signal Processing Letters*, v. 11, n. 2 PART II, p. 270 – 273, 2004. ISSN 10709908.

HU, Y.; LOIZOU, P. C. Speech enhancement based on wavelet thresholding the multitaper spectrum. *IEEE Transactions on Speech and Audio Processing*, v. 12, n. 1, p. 59 – 67, 2004. ISSN 10636676.

HU, Y.; LOIZOU, P. C. Subjective comparison and evaluation of speech enhancement algorithms. In: . [S.l.: s.n.], 2006. v. 1.

ITU-T. Objective measurement of active speech level. In: *ITU-T Recommendation P.56*. [S.l.: s.n.], 1993.

ITU-T. Subjective performance assessment of telephone-band and wideband digital codecs. In: *ITU-T Recommendation P.830*. [S.l.: s.n.], 1996.

ITU-T. Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. In: *ITU-T Recommendation P.862*. [S.l.: s.n.], 2001.

ITU-T. Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm. In: *ITU-T Recommendation P.835*. [S.l.: s.n.], 2003.

KATES, J. M.; AREHART, K. H. Coherence and the speech intelligibility index. *The Journal of the Acoustical Society of America*, ASA, v. 117, n. 4, p. 2224–2237, 2005. Disponível em: <http://link.aip.org/link-/?JAS/117/2224/1>.

KAY, S. M. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. 1. ed. [S.l.]: Prentice Hall, 1993. ISBN 0133457117.

KJEMS, U. et al. Role of mask pattern in intelligibility of ideal binary-masked noisy speech. *Journal of the Acoustical Society of America*, v. 126, n. 3, p. 1415 – 1426, 2009. ISSN 00014966.

LIM, J.; OPPENHEIM, A. All-pole modeling of degraded speech. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, v. 26, n. 3, p. 197–210, Jun 1978. ISSN 0096-3518.

LIM, J. S.; OPPENHEIM, A. V. Enhancement and bandwidth compression of noisy speech. *Proceedings of the IEEE*, IEEE, v. 67, n. 12, 1979.

LOIZOU, P.; KIM, G. Reasons why current speech-enhancement algorithms do not improve speech intelligibility and suggested solutions. *Audio, Speech, and Language Processing, IEEE Transactions on*, PP, n. 99, p. 1 –1, 2010. ISSN 1558-7916.

LOIZOU, P. C. *Speech Enhancement: Theory and Practice*. [S.l.]: CRC Press, 2007.

LU, C.-T. Reduction of musical residual noise for speech enhancement using masking properties and optimal smoothing. *Pattern Recognition Letters*, v. 28, n. 11, p. 1300 – 1306, 2007. ISSN 01678655.

MA, J.; HU, Y.; LOIZOU, P. C. Objective measures for predicting speech intelligibility in noisy conditions based on new band-importance functions. *Journal of the Acoustical Society of America*, v. 125, n. 5, p. 3387 – 3405, 2009. ISSN 00014966.

MONTGOMERY, D. C.; RUNGER, G. C. *Applied Statistics and Probability for Engineers*. Third edition. [S.l.]: John Wiley & Sons, Inc., 2003.

PAPOULIS, A. *Probability, Random Variables and Stochastic Processes*. 3rd edition. ed. [S.l.]: McGraw-Hill, 1991.

PLACK, C. J. *The Sense of Hearing*. New Jersey: Lawrence Erlbaum, 2005. ISBN 0805848843.

PLAPOUS, C.; MARRO, C.; SCALART, P. Speech enhancement using harmonic regeneration. In: . [S.l.: s.n.], 2005. I, p. I157 – I160. ISSN 15206149.

QUATIERI, T. *Discrete-Time Speech Signal Processing: Principles and Practice*. [S.l.]: Prentice Hall, 2002.

QUATIERI, T.; BAXTER, R. Noise reduction based on spectral change. In: . New Paltz, NY, USA: [s.n.], 1997. p. IEEE –.

QUATIERI, T.; DUNN, R. Speech enhancement based on auditory spectral change. In: . Orlando, FL, United states: [s.n.], 2002. v. 1, p. I/257 – I/260. ISSN 07367791.

SAID, A. White and color noise cancellation using adaptive feedback cross-coupled line enhancer filter. *8th IEEE International Symposium on Signal Processing and Information Technology, ISSPIT*, p. 96 – 99, December 2008.

SANTOS, S.; ALCAIM, A. Inventário reduzido de unidades fonéticas do português brasileiro para o reconhecimento de voz contínua. In: *SBT-97*. [S.l.: s.n.], 1997.

SCALART, P.; FILHO, J. V. Speech enhancement based on a priori signal to noise estimation. In: . Atlanta, GA, USA: [s.n.], 1996. v. 2, p. 629 – 632. ISSN 07367791.

SILVA, T. F. G. *Metodologia para a Aquisição de Sinais a Serem Utilizados em Simulações de Microfonia Direcional Adaptativa para Aparelhos Auditivos*. [S.l.], 2010.

SOHN, J.; KIM, N.; SUNG, W. Statistical model-based voice activity detection. *IEEE Signal Processing Letters*, Piscataway, NJ, United States, v. 6, n. 1, p. 1 – 3, 1999. ISSN 10709908.

SREENIVAS, T. V.; KIRNAPURE, P. Codebook constrained wiener filtering for speech enhancement. *IEEE Trans. Speech Audio Process*, v. 4, p. 383–389, 1996.

SUN, X. Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio. In: *Proc. of ICASSP*. [S.l.: s.n.], 2002. p. 200–2.

WIENER, N. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. [S.l.]: The MIT Press, 1964. ISBN 0262730057.

ANEXO A – PROGRAMAS UTILIZADOS

Algoritmo A.1: Algoritmo de Wiener Proposto

```
function wiener_as_proposed3_realVAD(filename,
    outfile, L_align,max_att,mult)

%
%  Implements the Wiener filtering algorithm based
    on a priori SNR estimation [1].
%
%  Usage:  wiener_as(noisyFile, outputFile)
%
%         infile - noisy speech file in .wav format
%         outputFile - enhanced output file in .wav
    format

%
%  Example call:  wiener_as('sp04_babble_sn10.wav
    ','out_wien_as.wav');
%
%  References:
%   [1] Scalart, P. and Filho, J. (1996). Speech
    enhancement based on a priori
%       signal to noise estimation. Proc. IEEE Int.
    Conf. Acoust. , Speech, Signal
%       Processing, 629-632.
%
% Authors: Yi Hu and Philipos C. Loizou
%
% Copyright (c) 2006 by Philipos C. Loizou
```

```
% $Revision: 0.0 $  $Date: 10/09/2006 $
%
    ------------------------------------------------------------


if nargin<2
   fprintf('Usage: wiener_as(noisyfile.wav,outFile.
       wav) \n\n');
   return;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[noisy_speech, fs, nbits]= wavread( filename);

% set parameter values


last_harm = false;

beta = 0.002;
mu= 0.98; % smoothing factor in noise spectrum
    update
a_dd= 0.98; % smoothing factor in priori update
eta= 0.18; % VAD threshold %%original
%eta= 10.0; % VAD threshold
frame_dur= 40; % frame duration (ms)
L= frame_dur* fs/ 1000; % L is frame length (160
    for 8k sampling rate)
hamming_win= hamming( L); % hamming window
U= ( hamming_win'* hamming_win)/ L; % normalization
     factor
```

```
% first 120 ms is noise only
len_120ms= fs/ 1000* 120;
% first_120ms= noisy_speech( 1: len_120ms).* ...
%     (hann( len_120ms, 'periodic'))';
first_120ms= noisy_speech( 1: len_120ms);

%max_att = 1.0;
min_att_def = 0.5;
min_att = min_att_def;
min_att_prev = max_att;
%mult = 3;

% =============now use Welch's method to estimate
    power spectrum with
% Hamming window and 50% overlap
nsubframes= floor( len_120ms/ (L/ 2))- 1;  % 50%
    overlap
noise_ps= zeros( L, 1);
n_start= 1;

for j= 1: nsubframes
    noise= first_120ms( n_start: n_start+ L- 1);
    noise= noise.* hamming_win;
    noise_fft= fft( noise, L);
    noise_ps= noise_ps+ ( abs( noise_fft).^ 2)/ (L*
        U);
    %noise_ps = noise_ps / max(noise_ps); %
        normalize vector
    n_start= n_start+ L/ 2;
end
noise_ps= noise_ps/ nsubframes;
```

```
%==============

% number of noisy speech frames
len1= L/ 2; % with 50% overlap
nframes= floor( length( noisy_speech)/ len1)- 1;
n_start= 1;

vad_decision = zeros(nframes,1);
for j= 1: nframes
    noisy= noisy_speech( n_start: n_start+ L- 1);
    noisy = noisy.* hamming_win;

    noisy_fft = fft( noisy, L);
    noisy_ps = ( abs( noisy_fft).^ 2)/ (L* U);


    min_att = 0.3*min_att + 0.7*min_att_prev;
    min_att_prev = min_att;

    if (j== 1) % initialize posteri
        posteri= noisy_ps./ noise_ps;
        posteri_prime= posteri- 1;
        posteri_prime( posteri_prime< 0)= 0;
        priori= a_dd+ (1-a_dd)* posteri_prime;

    else
        posteri= noisy_ps./ noise_ps;
        posteri_prime= posteri- 1;
        posteri_prime( posteri_prime< 0)= 0;
        priori= a_dd* (x_hat_ps_prev)./
            noise_ps_prev + ...
            (1-a_dd)* posteri_prime;
```

```
        end


111     % ============ voice activity detection pg 544
           Loizou book

        sigma_k = (1./(1+ priori)) .* exp(posteri.*
            priori./ (1+ priori));
        vad_decision(j)= sum( log10(sigma_k))/ L;
        if (vad_decision(j)< eta)
            % noise only frame found
            noise_ps= mu* noise_ps+ (1- mu)* noisy_ps;
            min_att = max_att;
            vad( n_start: n_start+ L- 1)= 0;
        else
121         vad( n_start: n_start+ L- 1)= 1;
            min_att = min_att_def;
        end
        % ===end of vad ===

        %%%%%%%%%%  Pitch Detection for each Frame
            %%%%%%%%%%%%%%
        [~,f0_value,~,~] = shrp(noisy,fs,[50 550],
            frame_dur);

        beta_norm = beta_gaussian_multiple2(f0_value, L
            , 245, fs, mult, last_harm,max_att,min_att)
            ';

131     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
        G= ( priori./ (1+ priori)).^beta_norm; % gain
            function

        x_hat = noisy_fft.* G;
        x_hat_ps = abs(x_hat).^2/(L*U);

        enhanced= ifft( x_hat, L);

        if ( j==1 )
141         enhanced_speech( n_start: n_start+ L/2- 1 )
                = ...
                 enhanced( 1: L/2);
        else
            enhanced_speech( n_start: n_start+ L/2- 1 )
                = ...
                 overlap+ enhanced( 1: L/2);
        end

        overlap= enhanced( L/2+ 1: L );
        n_start= n_start+ L/2;

151     noise_ps_prev = noise_ps;
        x_hat_ps_prev = x_hat_ps;

    end

    enhanced_speech( n_start: n_start+ L/2- 1)= overlap
        ;

    if (L_align == true)
        run_enh_LA(enhanced_speech,fs, nbits); %run
            level ajustment on enhanced signal
```

```
end

max_mod = max( abs( enhanced_speech));

if max_mod >= 1
    enhanced_speech = 0.99*(enhanced_speech/max_mod
        );
end

wavwrite( enhanced_speech, fs, nbits, outfile);
```

Algoritmo A.2: Função para gerar $\beta(\omega_k)$

```
function fun = beta_gaussian_multiple2(pitch, L,
    sigma, fs, mult,last_harmonic , max_att,
    min_att)

if nargin < 7
    max_att = 0.5;
    min_att = 0.1;
end
if nargin < 8
    min_att = 0.1;
9 end

L = L/2;
fs = fs/2;


f = zeros(mult,L);
for k=1:(mult)
    f(k,:) = beta_gaussian(pitch*k, L, sigma, fs,1,
```

```
            0); % generate only positive part of Beta
    end
19  fun = 1;
    for k = 1:mult
       fun = f(k,:).* fun; % generate only positive
           part of Beta
    end

    if (last_harmonic)
        last_h = (fix(fs/pitch-1))*pitch;
        fun = fun .* beta_gaussian(last_h, L, sigma, fs
            , 1, 0);

    end

    fun = (fun*(max_att-min_att) + min_att);

    fun = simetricBeta(fun); % generate full Beta

    end

    function fun =  simetricBeta(fun)

    fun = [fun fliplr(fun)];

    end


    function fun = beta_gaussian(pitch, L, sigma, fs,
        max_att, min_att)

    if nargin < 5
```

```
        max_att = 0.5;
        min_att = 0.1;
    end
    if nargin < 6
        min_att = 0.1;
    end

    X = linspace(0,fs,L); %generates a vector X of L
        points linearly spaced
                              %between and including 0
                                  and fs

    mu = pitch; %frequency [Hz]

    pico = 1/(sqrt(2*pi*sigma^2));
    Y = normpdf(X,mu,sigma);
    fun = (-Y + pico);
    fun = (max_att-min_att) * fun /(pico) + min_att;

    end
```

Algoritmo A.3: Função para gerar gráficos de barras

```
% option for test parameter
% if test = 'pesq' -> run stats for pesq
% if test = 'csii' -> run stats for csii
% if test = 'csii' -> run stats for csii
function plotTtest_final(test,n_val,vad_choice,att)

 %vad_choice = 1; %-> VAD Ideal
 %vad_choice = 2; %-> VAD Real
```

```
mainPath = 'D:\Mestrado UFSC\Trim 3\trabalho
    orientado\single microphone final\Signals for
    stat Tests\';

noise_opt = {'white_noise','cooler_noise','
    babble_noise',...
                    'vaccum_cleaner_noise','
                        train_noise'};
noise_type = noise_opt(n_val);

%gender = 1 -> female
%gender = 2 -> male
%gender = 1 -> male and female

for gender = 1:3

    plotTtest_( att, mainPath, test, cell2mat(
        noise_type), gender,vad_choice ); %ok

end

end

function plotTtest_(att,mainPath,test,noise_type,
    gender,vad_choice)



if (gender == 1)
    g_cmp = 'F';
    suffix = '_fem';
elseif gender == 2
```

```
        g_cmp = 'M';
37      suffix = '_male';
    else
        suffix = '_all';
    end

    if vad_choice == 1
        vad_x = '_ideal';
        enhanced_path = [mainPath,'enhanced signals\',
            noise_type,'\',test,' scores\'];
        enhanced_path_ = [enhanced_path,test,'_stats',
            suffix,'_',num2str(att),'.txt'];
    elseif vad_choice == 2
47      vad_x = '_real';
        enhanced_path = [mainPath,'enhanced signals\
            real VAD\',noise_type,'\',test,' scores\'];
        enhanced_path_ = [enhanced_path,test,'_stats',
            suffix,'_',num2str(att),'.txt'];
    else
        vad_x = '_noiseEst';
        enhanced_path = [mainPath,'enhanced signals\
            noise Estimation\',noise_type,'\',test,'
            scores\'];
        enhanced_path_ = [enhanced_path,test,'_stats',
            suffix,'_',num2str(att),'.txt'];
    end


57  %open file
    fid = fopen(enhanced_path_,'r');
    %Read the first line of the file containing the
        header information
```

```
headerline = fgetl(fid);
%The file pointer is now at the beginning of the
    second line. Use TEXTSCAN to read the columns
    of data.
data = textscan(fid,'%s%f%f%f%f%f%f%s%s%s','
    Delimiter',',');
%close file
fclose(fid);

%get values into variables
snr_ = data{1};
stats_mean = [data{2} data{3}];
stats_std = [data{4} data{5}];
ttest_val = data{7};

% convert data do desired format
snr_cell = set_snrCell(snr_);
signs = convert_toSigns(ttest_val);

%ttle_nse = cell2mat(noise_type);
ttle_nse = regexprep(noise_type, '_', ' ');
title_p = {['T-test for female voices - ',ttle_nse
    ],...
           ['T-test for male voices - ',ttle_nse
               ],...
           ['T-test for male and female voices - ',
               ttle_nse]};

if strcmp( test, 'csii')
    ttle = 'CSII';
else ttle = 'PESQ';
end
```

```
87 h = figure();
   handles = barweb(stats_mean, stats_std, 1, snr_cell
       , ...
        cell2mat(title_p(gender)) , 'SNR', ttle,...
        'bone', 'none', {'Conventional algorithm', '
            Proposed algorithm'}, 2, 'plot',signs);
   aux_name = title_p;
   aux_name = regexprep(aux_name, '-', '_');
   aux_name = regexprep(aux_name, ' ', '_');
   image_name = [cell2mat(aux_name(gender)),ttle,vad_x
       ,'.eps'];
   print(h, '-dpsc', ['C:\Users\luiz\Desktop\
       noise_est_imcra\',image_name]);
   end

   function snr_cell = set_snrCell(snr_)

   for k = 1:size(snr_)

        snr_cell{k} = [char(snr_(k)), ' dB'];

   end

   end

   function signs = convert_toSigns(ttest_val)

   signs = zeros(size(ttest_val));
   signs = num2str(signs);

   for k = 1:size(ttest_val)
```

```
        if( ttest_val(k) ~= 0)
            signs(k) = '*';
        else
117         signs(k) = ' ';
        end
    end

    end

    function handles = barweb(barvalues, errors, width,
        groupnames, bw_title, bw_xlabel, bw_ylabel,
        bw_colormap, gridstatus, bw_legend, error_sides
        , legend_type, signs)

    %
    % Usage: handles = barweb(barvalues, errors, width,
        groupnames, bw_title, bw_xlabel, bw_ylabel,
        bw_colormap, gridstatus, bw_legend, error_sides
        , legend_type, signs)
127 %
    % Ex: handles = barweb(my_barvalues, my_errors, [],
        [], [], [], [], bone, [], bw_legend, 1, 'axis
        ')
    %
    % barweb is the m-by-n matrix of barvalues to be
        plotted.
    % barweb calls the MATLAB bar function and plots m
        groups of n bars using the width and
        bw_colormap parameters.
    % If you want all the bars to be the same color,
        then set bw_colormap equal to the RBG matrix
        value ie. (bw_colormap = [1 0 0] for all red
```

```
    bars)
% barweb then calls the MATLAB errorbar function to
    draw barvalues with error bars of length error
    .
% groupnames is an m-length cellstr vector of
    groupnames (i.e. groupnames = {'group 1'; '
    group 2'}).  For no groupnames, enter [] or {}
% The errors matrix is of the same form of the
    barvalues matrix, namely m group of n errors.
% Gridstatus is either 'x','xy', 'y', or 'none' for
    no grid.
% No legend will be shown if the legend paramter is
    not provided
% 'error_sides = 2' plots +/- std while '
    error_sides = 1' plots just + std
% legend_type = 'axis' produces the legend along
    the x-axis while legend_type = 'plot' produces
    the standard legend.  See figure for more
    details
%
% The following default values are used if
    parameters are left out or skipped by using [].
% width = 1 (0 < width < 1; widths greater than 1
    will produce overlapping bars)
% groupnames = '1', '2', ... number_of_groups
% bw_title, bw_xlabel, bw_ylabel = []
% bw_color_map = jet
% gridstatus = 'none'
% bw_legend = []
% error_sides = 2;
% legend_type = 'plot';
%
```

```
% A list of handles are returned so that the user
    can change the properties of the plot
% handles.ax: handle to current axis
% handles.bars: handle to bar plot
% handles.errors: a vector of handles to the error
    plots, with each handle corresponding to a
    column in the error matrix
% handles.legend: handle to legend
%
%
% See the MATLAB functions bar and errorbar for
    more information
%
% Author: Bolu Ajiboye
% Created: October 18, 2005 (ver 1.0)
% Updated: Dec 07, 2006 (ver 2.1)
% Updated: July 21, 2008 (ver 2.3)
% Modified by Luiz F. Silva - Jan, 2011

% Get function arguments
if nargin < 2
  error('Must have at least the first two arguments
      :  barweb(barvalues, errors, width,
      groupnames, bw_title, bw_xlabel, bw_ylabel,
      bw_colormap, gridstatus, bw_legend,
      barwebtype)');
elseif nargin == 2
  width = 1;
  groupnames = 1:size(barvalues,1);
  bw_title = [];
  bw_xlabel = [];
  bw_ylabel = [];
```

```
      bw_colormap = jet;
      gridstatus = 'none';
177   bw_legend = [];
      error_sides = 2;
      legend_type = 'plot';
    elseif nargin == 3
      groupnames = 1:size(barvalues,1);
      bw_title = [];
      bw_xlabel = [];
      bw_ylabel = [];
      bw_colormap = jet;
      gridstatus = 'none';
187   bw_legend = [];
      error_sides = 2;
      legend_type = 'plot';
    elseif nargin == 4
      bw_title = [];
      bw_xlabel = [];
      bw_ylabel = [];
      bw_colormap = jet;
      gridstatus = 'none';
      bw_legend = [];
197   error_sides = 2;
      legend_type = 'plot';
    elseif nargin == 5
      bw_xlabel = [];
      bw_ylabel = [];
      bw_colormap = jet;
      gridstatus = 'none';
      bw_legend = [];
      error_sides = 2;
      legend_type = 'plot';
```

```
207 elseif nargin == 6
      bw_ylabel = [];
      bw_colormap = jet;
      gridstatus = 'none';
      bw_legend = [];
      error_sides = 2;
      legend_type = 'plot';
    elseif nargin == 7
      bw_colormap = jet;
      gridstatus = 'none';
217   bw_legend = [];
      error_sides = 2;
      legend_type = 'plot';
    elseif nargin == 8
      gridstatus = 'none';
      bw_legend = [];
      error_sides = 2;
      legend_type = 'plot';
    elseif nargin == 9
      bw_legend = [];
227   error_sides = 2;
      legend_type = 'plot';
    elseif nargin == 10
      error_sides = 2;
      legend_type = 'plot';
    elseif nargin == 11
      legend_type = 'plot';
    end

    change_axis = 0;
237 ymax = 0;
```

```
    if size(barvalues,1) ~= size(errors,1) || size(
        barvalues,2) ~= size(errors,2)
      error('barvalues and errors matrix must be of
          same dimension');
    else
      if size(barvalues,2) == 1
        barvalues = barvalues';
        errors = errors';
      end
      if size(barvalues,1) == 1
247     barvalues = [barvalues; zeros(1,length(
            barvalues))];
        errors = [errors; zeros(1,size(barvalues,2))];
        change_axis = 1;
      end
      numgroups = size(barvalues, 1); % number of
          groups
      numbars = size(barvalues, 2); % number of bars in
          a group
      if isempty(width)
        width = 1;
      end

257   % Plot bars
      handles.bars = bar(barvalues, width,'edgecolor','
          k', 'linewidth', 2);
      hold on
      if ~isempty(bw_colormap)
        colormap(bw_colormap);
      else
        colormap(jet);
      end
```

```
if ~isempty(bw_legend) && ~strcmp(legend_type, '
    axis')
  handles.legend = legend(bw_legend, 'location',
      'NorthEast', 'fontsize',14);
  legend boxoff;
else
  handles.legend = [];
end

% Plot erros
for i = 1:numbars
  x =get(get(handles.bars(i),'children'), 'xdata'
      );
  x = mean(x([1 3],:));
      handles.errors(i) = errorbar(x, barvalues
          (:,i), errors(:,i), 'Color',[.44 .5
          .56], 'linestyle', 'none', 'linewidth',
           2);
  ymax = max([ymax; barvalues(:,i)+errors(:,i)]);
end

if error_sides == 1
  set(gca,'children', flipud(get(gca,'children'))
      );
end

ylim([0 ymax*1.5]);
xlim([0.5 numgroups-change_axis+0.5]);

if strcmp(legend_type, 'axis')
  for i = 1:numbars
    xdata = get(handles.errors(i),'xdata');
```

```
        for j = 1:length(xdata)
          text(xdata(j),  -0.03*ymax*1.1, bw_legend(i
              ), 'Rotation', 60, 'fontsize', 20, '
              HorizontalAlignment', 'right');
        end
      end
      set(gca,'xaxislocation','top');
    end

297 if ~isempty(bw_title)
      title(bw_title, 'fontsize',18);
    end
    if ~isempty(bw_xlabel)
      xlabel(bw_xlabel, 'fontsize',18);
    end
    if ~isempty(bw_ylabel)
      ylabel(bw_ylabel, 'fontsize',18);
    end

307 set(gca, 'xticklabel', groupnames, 'box', 'off',
        'ticklength', [0 0],...
          'fontsize', 14, 'xtick',1:numgroups, '
              linewidth', 2,'xgrid','off','ygrid','
              off');
    if ~isempty(gridstatus) && any(gridstatus == 'x')
      set(gca,'xgrid','on');
    end
    if ~isempty(gridstatus) && any(gridstatus ==  'y'
        )
      set(gca,'ygrid','on');
    end
```

```
    handles.ax = gca;

      for i=1:length(barvalues)
         max_bvals = max(barvalues(i,:));
         max_evals = max(errors(i,:));
         sign_pos(i) = (max_bvals + max_evals)*1.1;
      end
      sign_pos = sign_pos';
      if (signs ~= 0)
      xdata = get(handles.errors(1),'xdata');
      for j = 1:length(xdata)
327     text(xdata(j) + width/4, sign_pos(j), signs(j
            ));
      end
      end

    hold off
  end
```

ANEXO B – DETALHES DE IMPLEMENTAÇÃO

A parte de implementação do trabalho e entendimento das normas para adequar os testes com uma metodologia reconhecida cientificamente foi a parte mais extensa do trabalho.

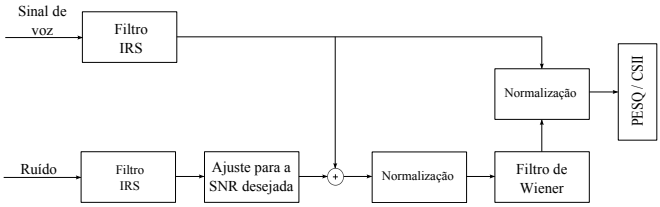


Figura 27: Diagrama de ilustração do procedimento de normalização e ajuste SNR

O algoritmo do PESQ já aplica automaticamente o filtro IRS aos sinais de entrada para simular o efeito de um canal de comunicação. No entanto, para que possamos ouvir o que está sendo processado pelo algoritmo devemos aplicar o filtro previamente conforme o diagrama acima. Além disso é importante que os sinais sejam normalizados utilizando o Nível de Fala Ativa de acordo com o método B especificado na norma (ITU-T, 1993) para que todos os sinais tenham suas potências ajustadas em relação ao sinal de voz. Esse método é utilizado tanto para ajustar a SNR desejada, quanto para normalizar os sinais após aplicarmos os algoritmo de redução de ruído. A Figura 27 ilustra esse procedimento na forma de diagrama de blocos para simplificar a compreensão.

Todos os algoritmos utilizados para a execução do presente trabalho foram programados em Matlab. Tanto os algoritmos de redução de ruído como também as ferramentas de análise estatística também foram utilizadas no Matlab.

Os algoritmos para obtenção do PESQ e CSII podem ser encontrados

no disco que acompanha o livro Speech Enchancement (LOIZOU, 2007). O algoritmo tradicional de Wiener também consta nesse disco. Esse algoritmo foi utilizado como base para o desenvolvimento do algoritmo proposto. O programa utiliza um VAD que é descrito matematicamente no próprio livro. No entanto, sua implementação apresenta um pequeno erro. Em uma das equações, é necessário utilizar o logaritmo de base dez e no Matlab é possível usá-lo através da função LOG10. Porém, o programa apresenta-se programado com a função LOG, que caracteriza logaritmo neperiano. O defeito pode ser facilmente corrigido trocando um função pela outra. Os algoritmos utilizados para esse trabalho foram corrigidos previamente para evitar conclusões equivocadas.

Os sinais de voz ruidosos foram processados pelos algoritmos utilizados neste trabalho e gravados para posterior análise. Para obtermos os índices PESQ e CSII utilizamos as respectivas rotinas em Matlab para cada indicador. O procedimento adotado foi:

1. Aplicar algoritmo PESQ aos sinais processados com os algoritmos AWC e AWP para cada um dos parâmetros existentes;

2. Aplicar algoritmo CSII aos sinais processados com os algoritmos AWC e AWP para cada um dos parâmetros existentes;

3. Gravar resultados dos indicadores em arquivos de texto no formato de valores separados por vírgula (CSV).

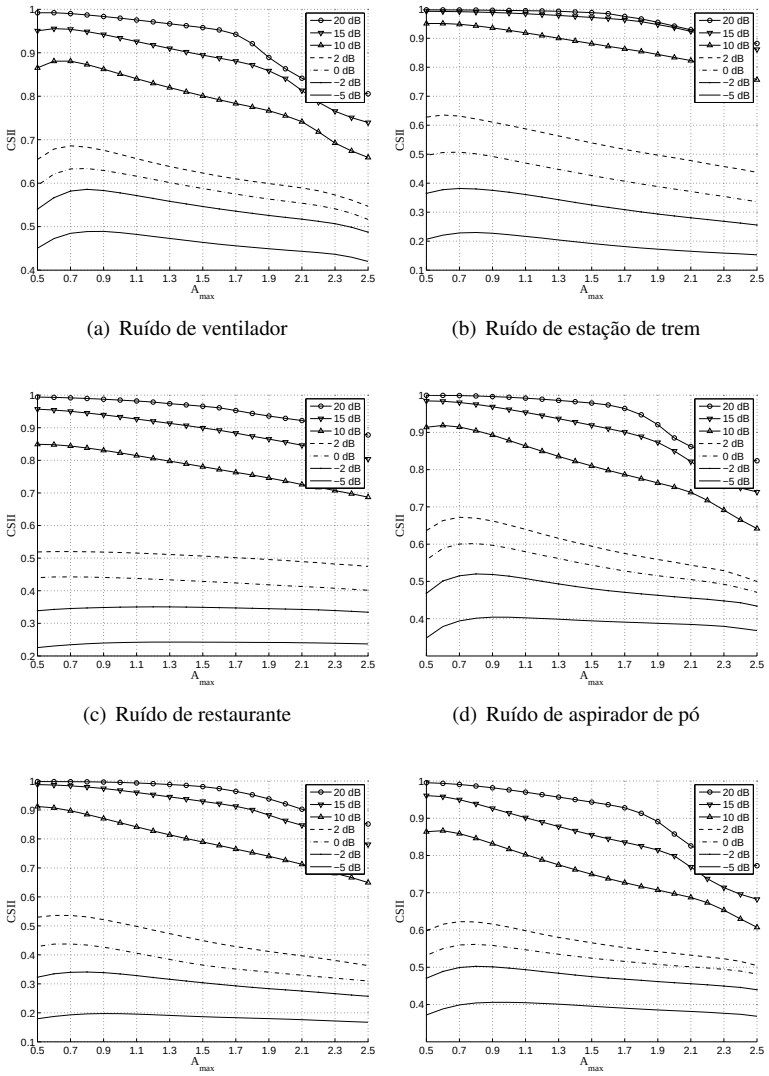
Após estes passos, foi utilizada uma nova rotina para ler os índices (PESQ e CSII) armazenados nos arquivos e então calcular a média, desvio padrão e também efetuar a análise estatística (teste-t) para cada um dos arquivos. Estes cálculos foram feitos tanto para o AWC quanto para o AWP. Os resultados foram armazenados em um novo arquivo de texto com as informações sobre os parâmetros dos algoritmos utilizados para o processamento

de cada sinal de voz. Isto facilitou identificação e utilização dos resultados, como por exemplo, plotar gráficos para posterior análise.

Os gráficos em forma de barras foram feitos de forma automática através de uma rotina também programada no ambiente Matlab. Com essa rotina foi possível gerar todos os gráficos e salvá-los automaticamente. Parte da rotina utilizada (*barweb*) foi originalmente programada por (AJIBOYE, 2006) e modificada para as necessidades do trabalho. A versão modificada encontra-se no Anexo A.

ANEXO C – COMPARAÇÃO DOS MELHORES CASOS

Nesta seção são mostrados os casos restantes para as curvas comparativas de $N = 3$ para várias SNRs. As Figuras 28 e 29 ilustram a comparação do efeito de $A_{\max}$ em voz masculina para diferentes SNRs utilizando os indicadores CSII e PESQ, respectivamente, para sinais de voz masculina.



(a) Ruído de ventilador

(b) Ruído de estação de trem

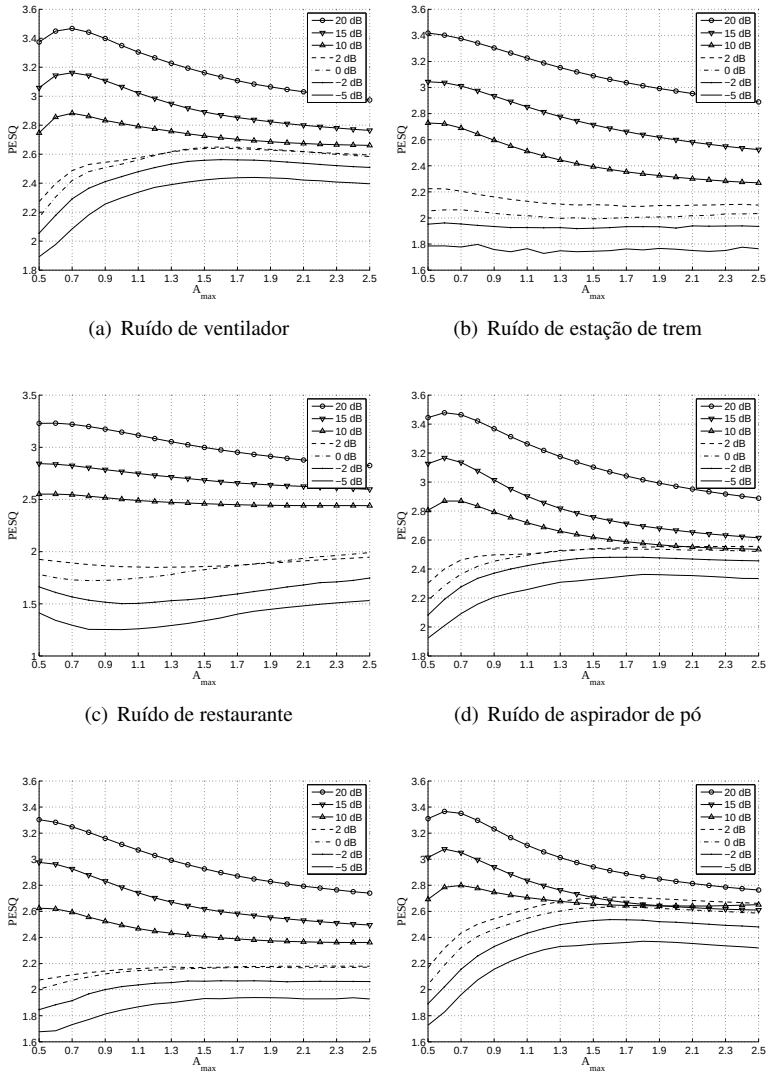
(c) Ruído de restaurante

(d) Ruído de aspirador de pó

(e) Ruído de rua movimentada

(f) Ruído branco

Figura 28: Comparação do efeito de $A_{\max}$ em voz masculina para diferentes SNRs, utilizando CSII.



(a) Ruído de ventilador

(b) Ruído de estação de trem

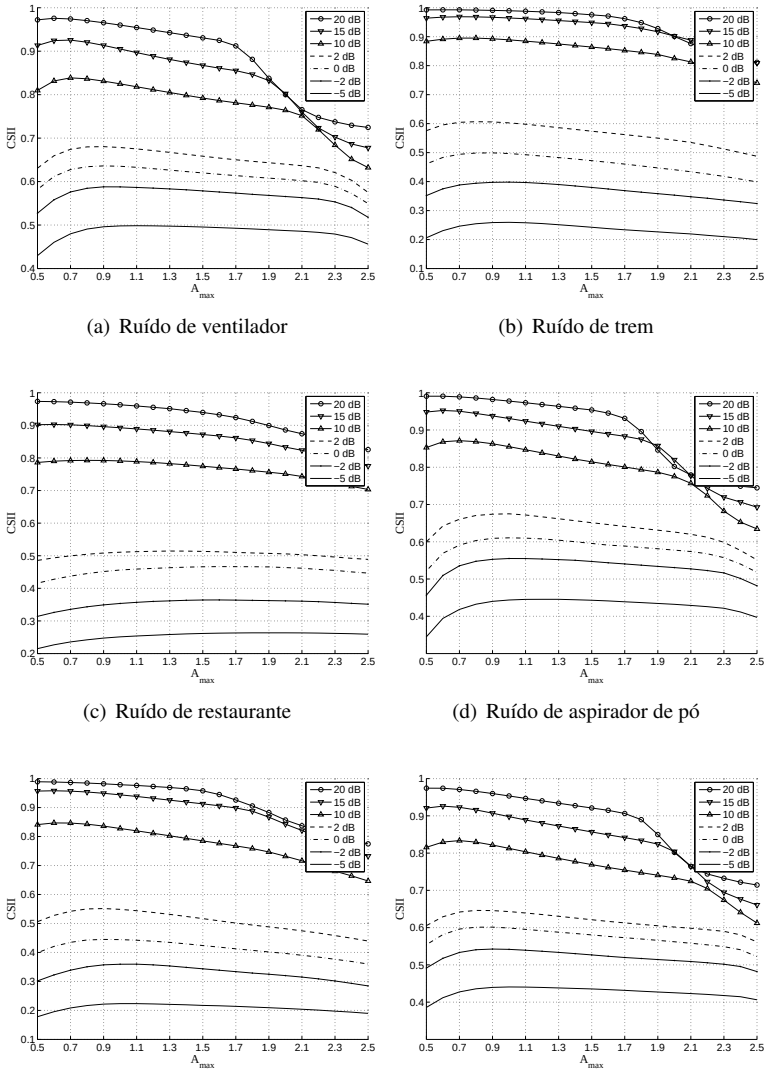
(c) Ruído de restaurante

(d) Ruído de aspirador de pó

(e) Ruído de rua movimentada

(f) Ruído branco

Figura 29: Comparação do efeito de $A_{\max}$ em voz masculina para diferentes SNRs, utilizando PESQ.



(a) Ruído de ventilador

(b) Ruído de trem

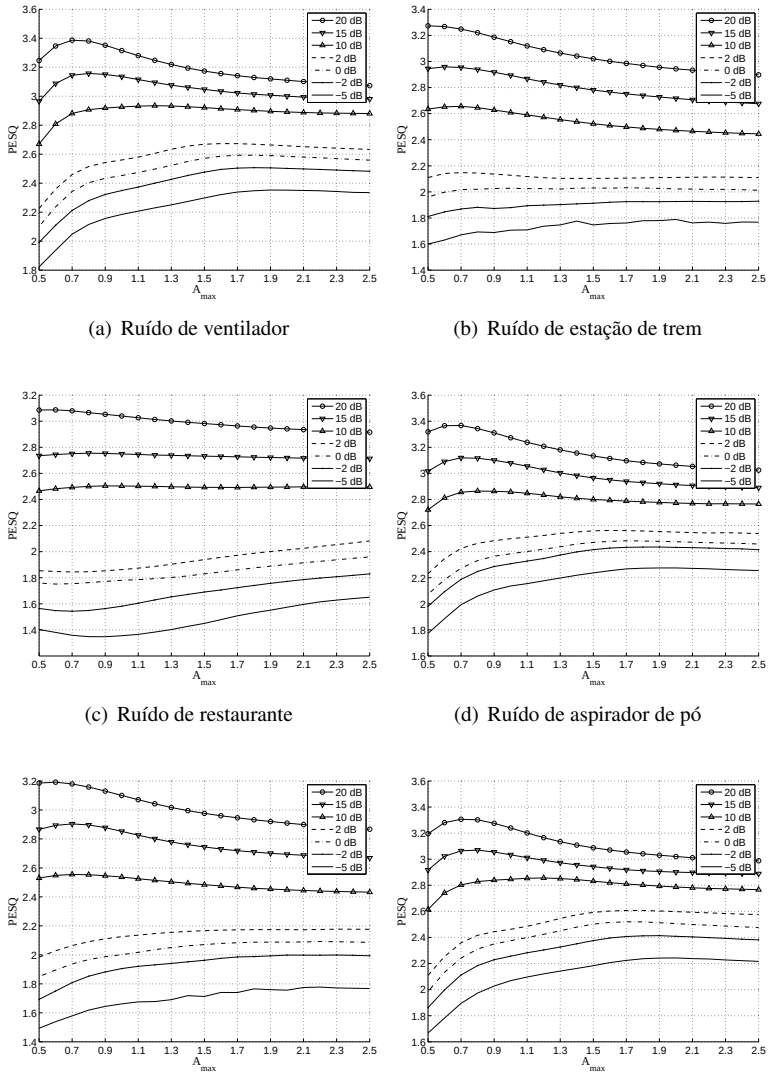
(c) Ruído de restaurante

(d) Ruído de aspirador de pó

(e) Ruído de rua movimentada

(f) Ruído branco

Figura 30: Comparação do efeito de $A_{\max}$ em voz feminina para diferentes SNRs, utilizando CSII.



(a) Ruído de ventilador

(b) Ruído de estação de trem

(c) Ruído de restaurante

(d) Ruído de aspirador de pó

(e) Ruído de rua movimentada

(f) Ruído branco

Figura 31: Comparação do efeito de $A_{\max}$ em voz feminina para diferentes SNRs, utilizando PESQ.

ANEXO D – SUPERFÍCIES DE DESEMPENHO

D.1 Ruído de aspirador de pó

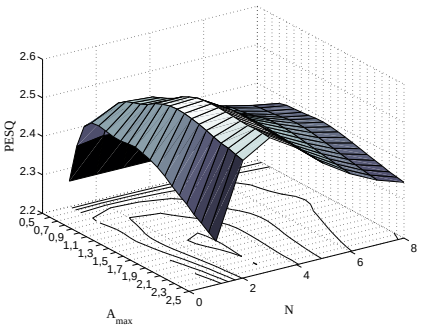
Sinais corrompidos por ruído de aspirador de pó é outro caso onde o algoritmo proposto se comporta da mesma maneira que o ruído branco. As Figuras 32(a) e 33(a) apresentam as vistas tri-dimensionais das superfícies de desempenho usando o PESQ e CSII, respectivamente. As Figuras 32(b) e 32(c) apresentam as vistas laterais. Estas vistas mostram claramente que temos um grande ganho em qualidade de voz quando comparado ao algoritmo tradicional [1]. Procurando pela linha $N = 3$ na Figura 33(c) (porque na Fig. 33(b) $A_{\max}$ é maximizado quando $N = 3$) nós podemos ver facilmente que o valor que maximiza a inteligibilidade é $A_{\max} \approx 0,8$ e, novamente, caímos na mesma relação de compromisso entre qualidade e inteligibilidade. É necessário, portanto, procurar por um valor ótimo para obter o melhor resultado possível em ambos os critérios. Baseado em testes subjetivos concluimos que esse valor é $A_{\max} \approx 0,8$.

Podemos notar também na Figura 12(d) que ao longo da linha $N = 3$ há diferentes intervalos de maximização da inteligibilidade para casa SNR considerada. Em SNRs altas há um intervalo estreito de ganho em inteligibilidade, por exemplo $A_{\max} \approx 0,8$, mas a medida que a SNR tende a valores menores, o intervalo de ganho em inteligibilidade é mais amplo e inicia em um valor de $A_{\max}$ um pouco maior, que é exatamente o mesmo que aconteceu no caso de ruído de ventilador.

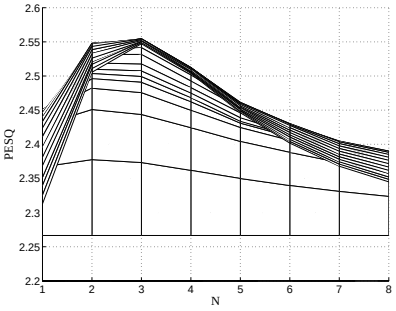
D.2 Ruído de estação de trem

Analisando os sinais corrompidos com ruído de estação de trem, vemos que as superfícies de desempenho são diferentes do ruído branco. As

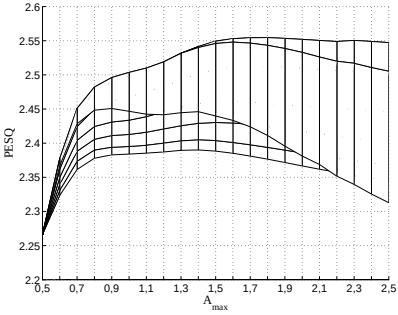
[1] Os valores do PESQ e CSII onde $A_{\max} = 0,5$ representam os valores atingidos pelo algoritmo tradicional e estão integrados às curvas para estabelecer uma comparação mais objetiva



(a) Vista 3D

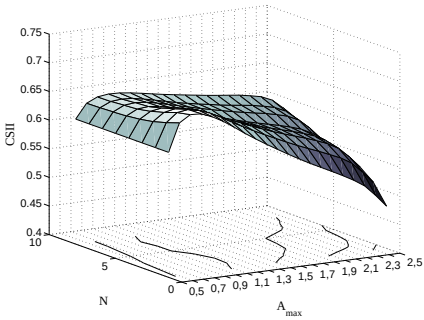


(b) Vista lateral direita

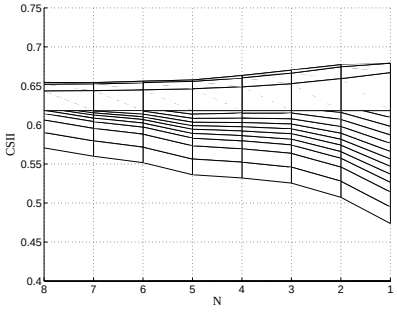


(c) Vista lateral esquerda

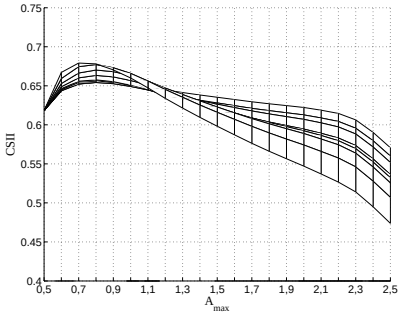
Figura 32: Superfícies de desempenho para ruído de aspirador de pó (SNR = 2 dB)



(a) Vista 3D



(b) Vista lateral direita



(c) Vista lateral esquerda

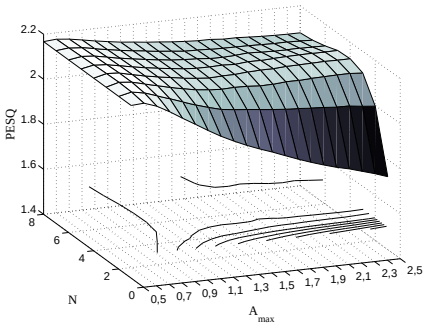
Figura 33: Superfícies de desempenho para ruído de aspirador de pó (SNR = 2 dB)

superfícies anteriores eram côncavas, o que significa que tínhamos um região de maximização bem definida. Agora, a superfície utilizando o PESQ é quase toda plana (exceto para $N = 1$ e $N = 2$) como pode ser visto na Figura 34(b). O valor de N não tem muita influência quando nos baseamos pela superfície do CSII (Figura 35(b)). Então a Figura 34(b) nos sevirá de referência e utilizaremos $N = 4$ como valor ótimo para o número de harmônicas. Isso significa que não há uma variação muito grande na qualidade do sinal de voz fora fora da região de maximização. Neste caso, os valores de maximização das superfícfies usando PESQ e CSII possuem o mesmo valor que é $A_{\max_o} = 0,7$. Isso pode ser verificado nas Figuras 34(c) e 35(c). As Figuras 34(a) e 35(a) mostram as vistas tri-dimensionais das superfícies utilizando os indicadores PESQ e CSII, respectivamente.

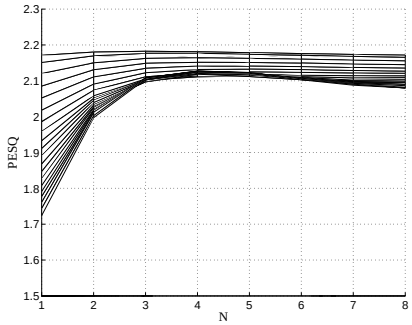
Embora os valores dos indicadores (PESQ e CSII) apontem um desempenho praticamente igual ao algoritmo convencional (lembre que $A_{\max} = 0,5$ é o valor obtido pelo AWC), testes subjetivos indicam uma melhora perceptível em inteligibilidade, bem como, redução do ruído musical.

D.3 Ruído de restaurante

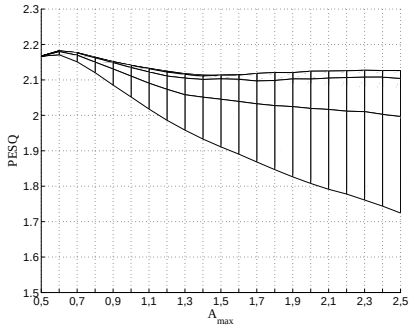
Podemos ver na Figura 36(a) que a superfície de desempenho obtida utilizando o indicador PESQ é bem diferente dos outros casos. Ao passo que a inteligibilidade teve uma boa melhora (Figura 37(c)), a qualidade subjetiva (Figura 36(c)) teve uma ligeira piora na região aceita como ideal nos casos anteriores. Nesse último gráfico é possível encontrar o ponto de máximo em $A_{\max} = 2,5$. Porém esse valor não reflete a realidade. Esse tipo de ruído aditivo é o pior cenário para o algoritmo de redução de ruído pois além de conter o barulho de talheres, também contém sinais de voz misturados. Devido à essas características, o algoritmo acaba cancelando também o sinal desejado. Quanto maior o valor de $A_{\max}$ empregado no algoritmo, mais o sinal desejado será corrompido, embora o indicador PESQ informe o contrário.



(a) Vista 3D

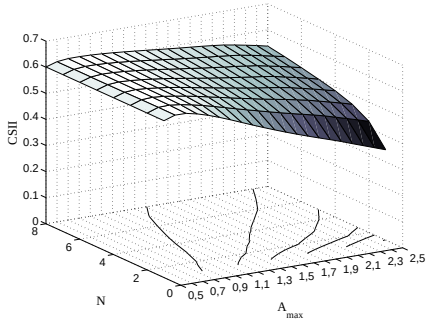


(b) Vista lateral direita

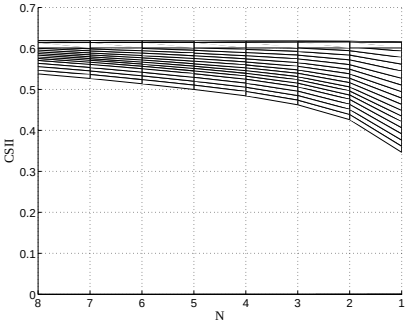


(c) Vista lateral esquerda

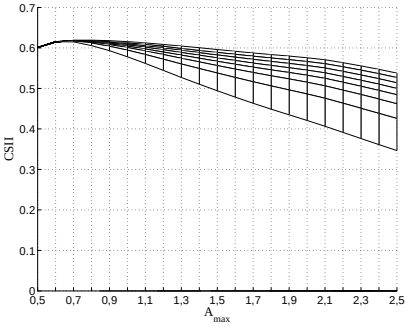
Figura 34: Superfícies de desempenho para ruído de estação de trem (SNR = 2 dB)



(a) Vista 3D



(b) Vista lateral direita

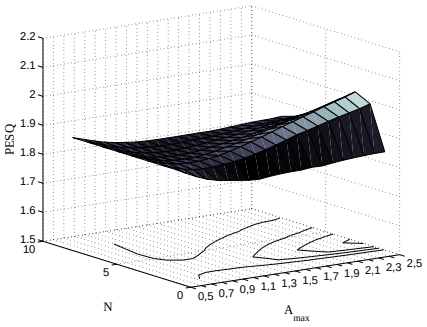


(c) Vista lateral esquerda

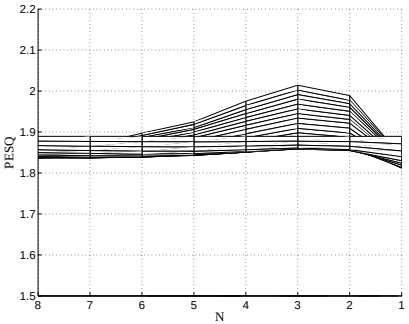
Figura 35: Superfícies de desempenho para ruído de estação de trem (SNR = 2 dB)

Desta forma, o PESQ servirá somente para escolhermos o número ótimo de harmônicas, $N = 3$. Para a escolha de $A_{\max_o}$ utilizaremos o indicador CSII.

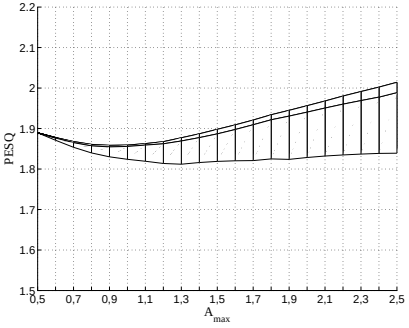
A Figura 37(a) mostra a vista tri-dimensional da superfície de desempenho para o indicador CSII. Na Figura 37(b) apresenta o gráfico para inteligibilidade. Este gráfico representa com mais exatidão o que o podemos perceber subjetivamente. O ponto máximo para o CSII encontra-se em torno de 1. Procurando pelo valor ótimo entre os dois critérios, escolhemos $A_{\max} \approx 1,0$ para fazer testes subjetivos, porque em torno deste valor a inteligibilidade é maximizada.



(a) Vista 3D

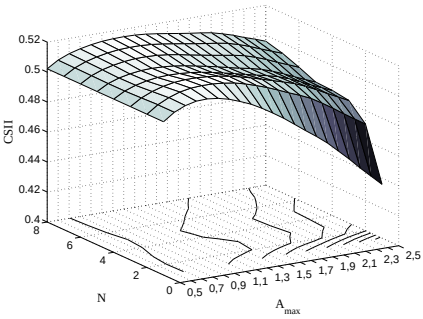


(b) Vista lateral direita

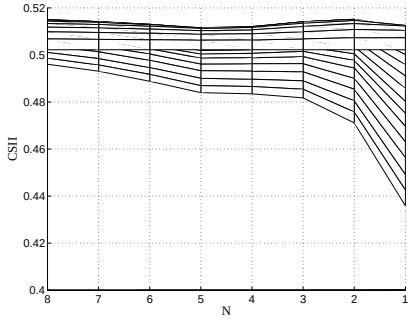


(c) Vista lateral esquerda

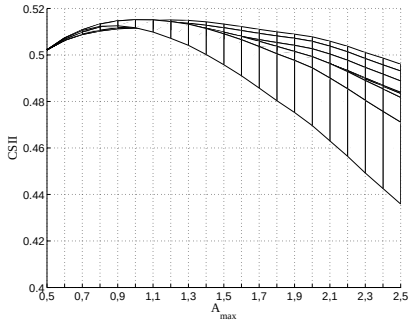
Figura 36: Superfícies de desempenho para ruído de restaurante (SNR = 2 dB)



(a) Vista 3D



(b) Vista lateral direita



(c) Vista lateral esquerda

Figura 37: Superfícies de desempenho para ruído de restaurante (SNR = 2 dB)